Single Cycle Instrument Placement

Report on the NASA Ames Milestone Demonstrations for the Intelligent Systems Program

September 30, 2004

Report prepared by

Liam Pedersen

QSS Group, Inc

NASA Ames Research Center

MS 269-3

Moffett Field, CA 94035-1000

Email: pedersen@email.arc.nasa.gov

Project Personnel and Contributors

Name	**Organization**	**Role**	**Contact**
Liam Pedersen, Ph.D	QSS Group, Inc	PI	pedersen@email.arc.nasa.gov
Srikanth Rajagopalan	QSS Group, Inc	PM	srikanth@email.arc.nasa.gov
Navigation, Instrument Placement and K9 Rover			
Randy Sargent	QSS Group, Inc	Instrument placement, mesh registration, Integration lead.	rsargent@email.arc.nasa.gov
Matthew Deans, Ph.D.	QSS Group, Inc	Keypoint tracker, vision	deano@email.arc.nasa.gov
Matt McLellan	Intern		
Ted Morse	Intern		
Maria Bualat	NASA ARC	K9	mbualat@mail.arc.nasa.gov
Clay Kunz	QSS Group, Inc	K9 Navigation and Software (CLARAty)	
Anne Wright	QSS Group, Inc	K9 Soft. Eng. (CLARAty)	
Eric Park	QSS Group, Inc	K9 Elec. Eng.	
Susan Lee	QSS Group, Inc	K9 Mech. Eng.	
Linda Kobayashi	NASA ARC	K9 Elec. Eng.	
Hoang Vu	NASA ARC	K9 Mech. Eng.	
Alan Chen	Intern	Instrument Placement	
Contingent Planning			
David E. Smith, Ph.D.	NASA ARC	Contingency Planning	De2smith@email.arc.nasa.gov

Ph.D.		Planning	
Nicolas Mealeau, Ph.D.	QSS Group, Inc	Monte-carlo estimator	nmeuleau@email.arc.nasa.gov
Sailesh Ramakrishnan	QSS Group, Inc	Planning	sailesh@email.arc.nasa.gov
Matthew Boyce	Intern	Planning	
Betty Lu	Intern	Planning	
CRL Executive			
Richard Washington, Ph.D.	Formerly RIACS	CRL Exec lead (retired)	
Howard Cannon	NASA ARC	CRL Exec lead	hcannon@email.arc.nasa.gov
Ray Garcia	QSS Group, Inc	CRL Exec lead (retired)	
Emmanuel Benazera	USRA-RIACS	Utility based plan re-evaluation	
Ground Data Systems			
David Lees, Ph.D.	QSS Group, Inc	Viz	lees@email.arc.nasa.gov
Larry Edwards, Ph.D.	NASA ARC	Viz	
Judd Bowman	Formerly QSS Group, Inc	Viz	
Leslie Keely	NASA ARC	Viz	
Ted Shab	QSS Group, Inc	PlanView	
Kim Hubbard	NASA ARC	PlanView	
Dennis Heher	QSS Group, Inc	Database	
Tom Dayton	QSS Group, Inc	PlanView User Experience	tdayton@email.arc.nasa.gov

		Design Leader	
Marleigh Norton	QSS Group, Inc	PlanView User Experience Design Intern	
Jay Trimble	NASA ARC	Merboard	Jay.P.Trimble@nasa.gov
Paul Backes, Ph.D.	JPL	Base Placement Engine lead	
Antonio Diaz-Calderon		Base placement engine Cognizant Engineer	
Mars Science			
Nathalie Cabrol, Ph.D	SETI Institute	Mars Science Consultant	ncabrol@mail.arc.nasa.gov
Gloria Hovde	SETI Institute	Science team coordinator	
Mission Simulation Facility (MSF)			
Greg Pisanich	QSS Group, Inc.	MSF project manager	gp@ptolemy.arc.nasa.gov
Lorenzo Flueckiger, Ph.D.	QSS Group, Inc.	MSF Lead Architect	lorenzo@email.arc.nasa.gov
Laura Plice	QSS Group, Inc.	MSF Systems Engineer	plice@ptolemy.arc.nasa.gov
Michael Wagner	QSS Group, Inc.	MSF-JPL Interface, ROAMS, VIZ	wagnermd@email.arc.nasa.gov
Chris Neukom PhD	QSS Group, Inc.	MSF-Robotic Systems	cneukom@mail.arc.nasa.gov
Eric Buchanan	Mission Critical Technologies	MSF Dynamics, Data collection	buchanan@email.arc.nasa.gov

Contents

Goals and Background

This project is building and integrating the diverse capabilities for an exploration rover to rapidly and reliably do multiple close-up and *in situ* contact measurements of objects in an unstructured and unpredictable environment, *with out* continuous operator supervision. This efficient goal level commanding capability represents an order of magnitude improvement in MER capabilities whilst requiring less operator support.

This research was motivated by the need of the planetary science community to acquire close up and contact measurements from a variety of targets on the surface of a planetary body. State-of-the-art planetary rovers, such as the MER rovers (Spirit and Opportunity) currently on Mars require 3 days and a standing army of operators on Earth to accomplish the task of driving up to a target and safely placing an instrument against it. With limited mission lifetimes and operations costs exceeding $1 million per day, decreasing this time and the number of operators has a significant scientific and cost-reduction pay-offs.

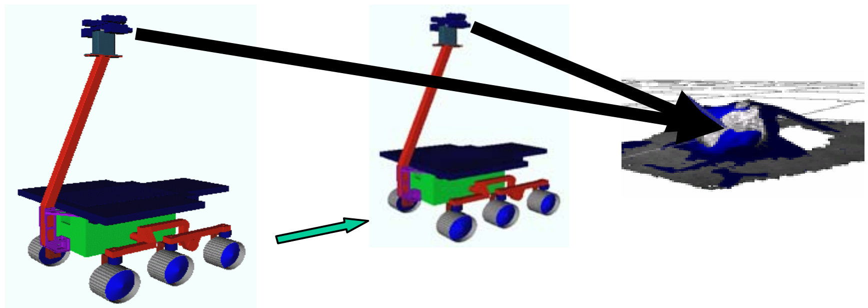
This project is building the capability for a rover to visit and examine multiple targets, scientific or otherwise, over 10's of meters in an un-prepared environment in one command cycle and without supervision from mission control. Using K9, a six wheeled planetary rover prototype, we have successfully demonstrated this in field locations, with operators at Ames communicating to it via satellite.

Research Overview

Achieving this has required us to make advances across a broad technological front:

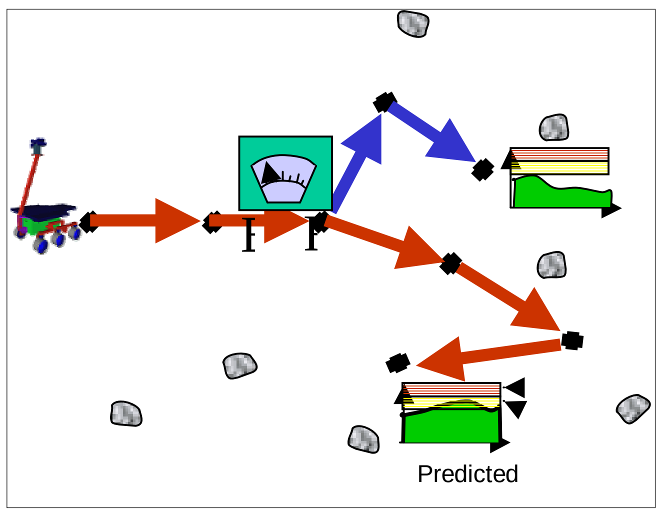
- *Target tracking and instrument placement* technologies to enable a rover to autonomously visit and examine many samples distributed over a 10m radius area with centimeter precision.

Because of wheel slippage and cumulative inertial guidance position errors, a rover cannot keep accurate track of goal locations around it using deduced reckoning alone as it moves towards them. Our solution has been the development of stereo-vision techniques using keypoints and 3D target templates to continuously track targets as the rover moves. No GPS or other infrastructure is required.



Once at the goal location, our auto-place algorithm permits the rover to distinguish rocks and other potential targets from the ground (regardless of slope or surface texture) and find instrument placements consistent with any limitations imposed by the tool and the target geometry.

- *Robust and flexible planning and execution* for the rover to accommodate the uncertainty associated with navigating to and deploying instruments on multiple samples, whilst adhering to the strict power, time and resource constraints characteristic of a planetary rover.



Standard mission practice is to generate daily activity plans off-board, permitting operators to modify and verify them prior to uplink. Whilst suitable for predictable systems, such as satellites in orbit, this approach copes poorly with uncertainty.

We have developed a ground based *contingency planner* that generates a main line rover activity sequence with flexible time constraints and contingent activity sequences to accommodate off-nominal behavior. These include diverting to closer targets if resource use is excessive and recovering from target tracking failures.

The rover *CRL Executive* executes these plans whilst monitoring resources and faults, and doing minor plan re-evaluations as required.

In addition, the CRL executive supports floating contingencies – activity sequences that can be activated at any point during execution in order to respond to unplanned for events, such as unforeseen science opportunities or rover endangering situations that require some immediate action.

This approach combines the benefits of the traditional approach with some of the flexibility but not the risk of an onboard planner.

- *Ground data systems* for users to rapidly identify, prioritize and specify many potential targets, evaluate the plan of action, and understand the data returned from the multiple samples the rover actually visited (which may differ from the highest priority set requested).

  Our operator interface uses the Viz software to immerse users in a photorealistic VR, 3D display of the environment around the rover. Within this, the users rapidly specify daily mission goals and evaluate returned data.

  Another tool, Merboard, facilitates collaboration amongst users and graphically displays forecast activities for and actual results from the rover.

- *Simulation-based Technology Development* The Mission Simulation Facility (MSF) provides a simulated testing environment including robotic vehicles, terrains, sensors, and vehicle subsystems. The MSF has been developed using a multi-platform distributed architecture that allows the simulation to be distributed across multiple machines and laboratories. Multi-platform support allows the MSF to easily integrate with existing simulation software developed on Unix, Linux, and Windows platforms.

  The MSF addresses several challenges often encountered by autonomy researchers. The first is access to the target platform. The MSF allows researchers to develop and test their software and algorithms before the real-world robotic platform is available as well as afterwards when the target platform is under heavy use.

Integrating autonomous systems (for example, planners, executives) is a second challenge. The MSF provides a mission-level environment where the scientists can integrate their individual modules, identifying and resolving incompatibilities. They can then quickly test the integration over many simulation runs, which would be time consuming using the target robotic platform. The figure here shows a simple demonstration mission that was used to test the integration of components.

Sample demo scenario in Mission Simulation Facility

The third is evaluating the resulting system prior to actual tests. The MSF is capable of replicating the target terrain, including the number and placement of rocks and obstacles. Software that performs well at this level of simulation has a higher probability of successfully achieving their goal.

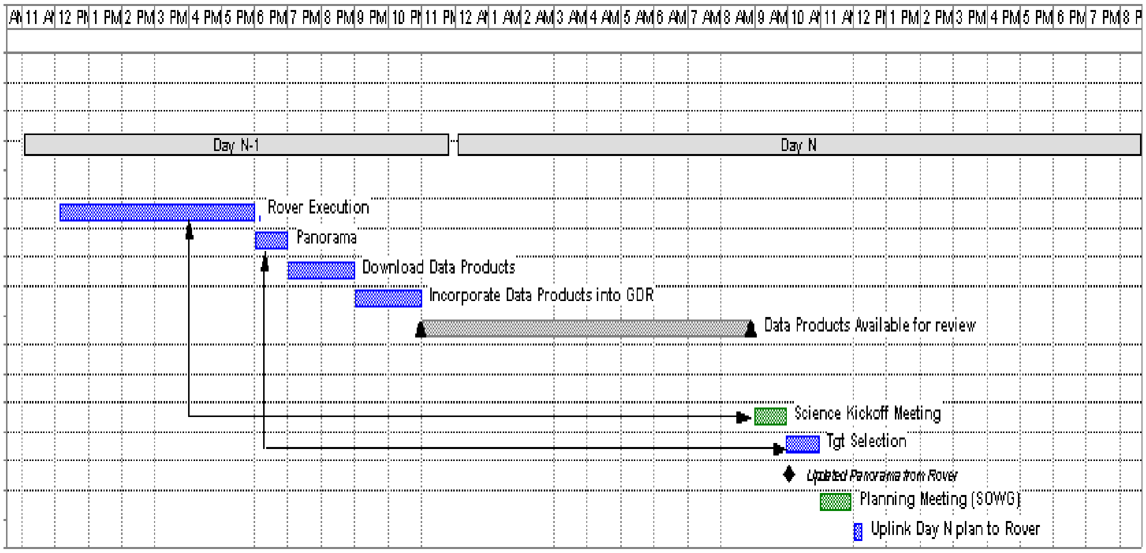
Table 1 : Technology Goals

<table>
<tr><td colspan="2">The goal of this technology driven project is to obtain close-up and contact measurements from an average of at least one, if not multiple science targets per command cycle from a single science rover</td></tr>
<tr><td>Technology</td><td>Supporting Goals</td></tr>
<tr><td>1. Fully autonomous navigation to targets and instrument placement</td><td>a. Autonomously track and navigate to science targets within local area, chosen by users
b. Autonomously place science instruments against rock targets, ensuring instrument and rover safety.</td></tr>
</table>

2. Contingent planning and robust execution for rover to adapt to increased uncertainty associated with autonomous navigation and instrument placement, whilst adhering to stringent resource (power and time) constraints.	a. Flexibly adds/remove science goals in response to changes in resource availability and usage (power, time). b. Obtain follow-up measurements to exploit new science opportunities discovered by on-board data analysis. c. Adapt science goals in response to basic faults (loss of target, inability to place instrument)
3. Effective ground data systems for users to interact with rover that operates for long durations under considerable uncertainty	a. Interface for users to express science goals b. Interface for users to plan/evaluate daily rover activities c. Enhance users situational awareness after complex activity plans with many uncertainties and variations d. Understand science user needs for interacting with highly autonomous systems
4. Simulation-based support for the development and testing of rover autonomy algorithms.	a. Provide a virtual rover running on virtual planetary terrain. b. Support software interfaces compatible with real rover data and commands. c. Provide the capability to test hypothetical scenarios including changes in terrain, uncertainty, and failure insertion.

Mission Scenario

Mission Story board

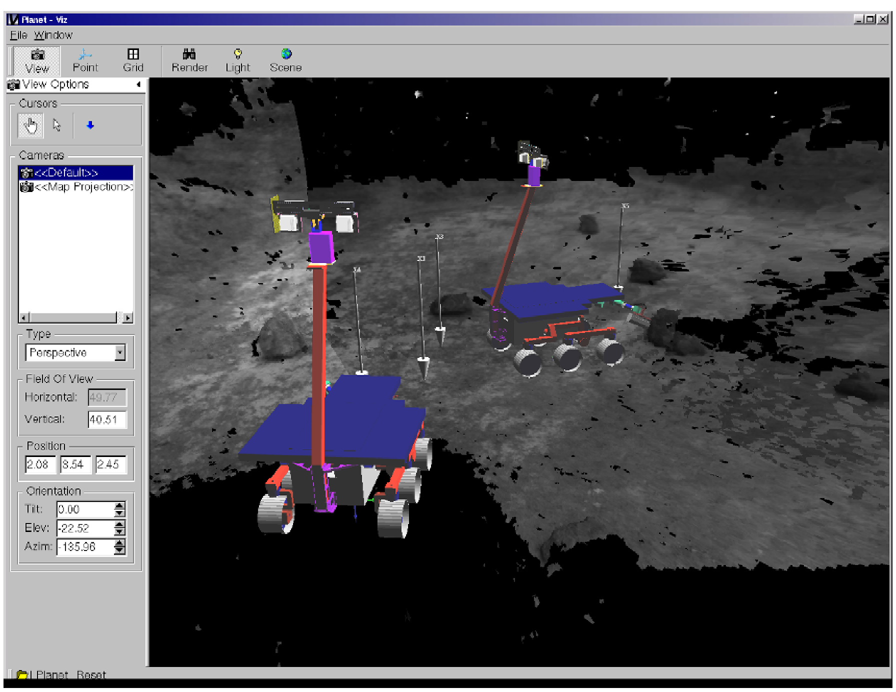


1) Science Kick-off Meeting:

   Mission scientists evaluate data products acquired by rover on previous sol to understand and confirm what measurements were acquired.

2) Target Selection:

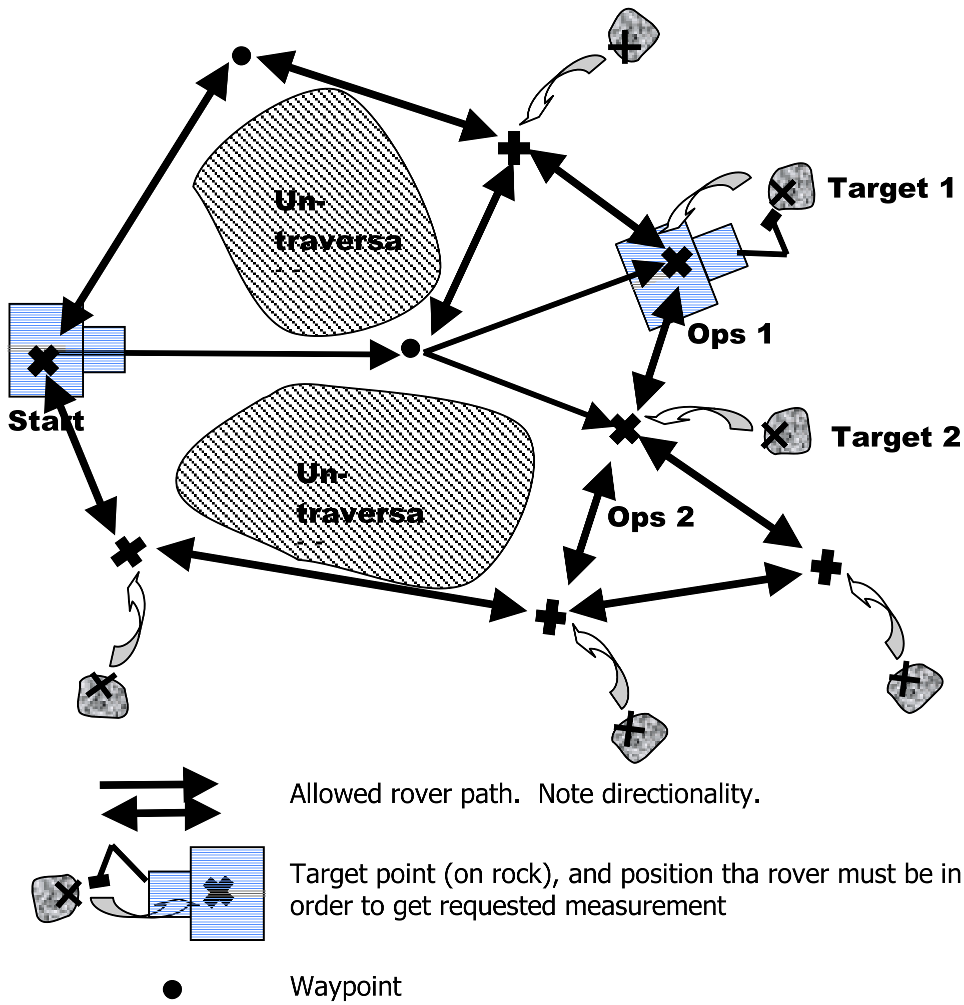
   Rover engineers generate photorealistic 3D virtual model of terrain around rovers current position (from recently uploaded panorama), and specify which areas are traversable by rover.



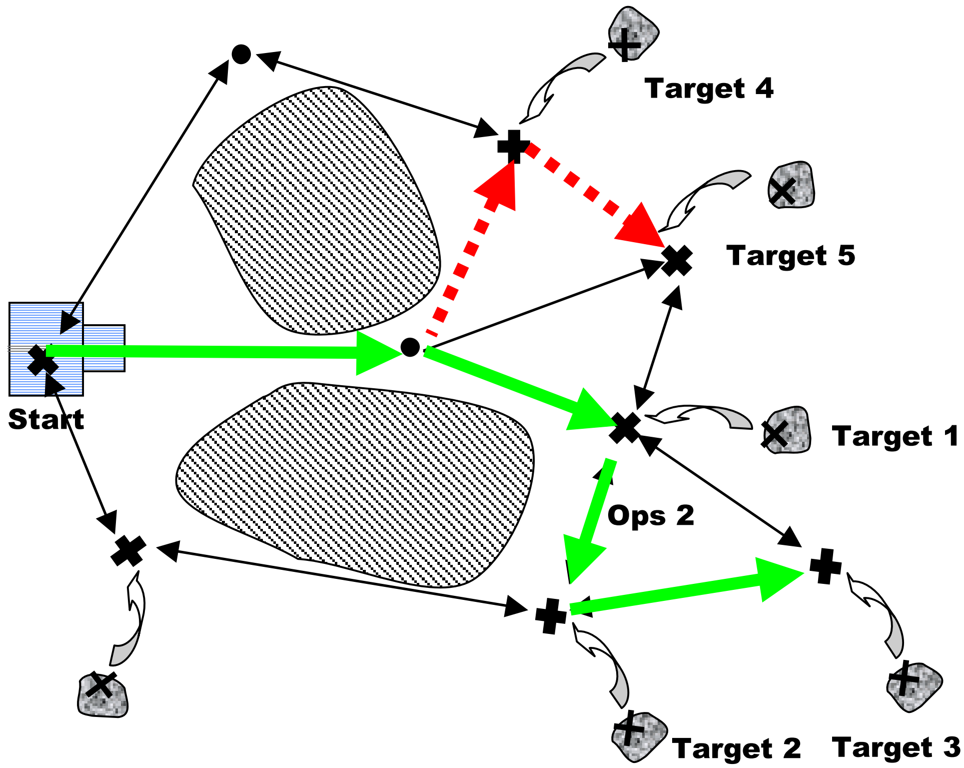
Mission scientists specify set of potential science targets in new model and desired measurements (close-up or contact) from each. These are prioritized according to their science value. Mission engineers, with input from the scientists, specify where the rover must be in order to acquire these measurements, as well as any additional constraints on the time of day.

3) Planning Meeting:

Mission engineers determine safe paths connecting rover start position and points from quire measurements of targets:



Mission engineers generate activity plan for rover to visit and get measurements from the most valuable subset of targets, subject to constraints on the total duration of rover activities, where it can go, and rover energy availability (green arrows):



The activity plan must be sufficiently flexible to cope with the relatively large uncertainty in time and energy required for the rover to navigate to and acquire measurements from so many targets. The plan may specify fall-back options (red arrows) for the rover to acquire other, less valuable, measurements in the event of faults, such as losing track of the primary targets, or insufficient resources and time remaining to complete the main plan (green arrows).

Mission engineers verify activity plan, and confirm with mission scientists that it is likely to accomplish desired science activities. Time permitting; the plan may be modified to accommodate scientist feedback.

4) Activity plan is uploaded to rover

5) Rover Execution of Activity Plan:

Rover navigates to desired locations to acquire measurements specified in plan. Whilst doing this, it keeps track of where the targets are, its remaining battery power, the time of day.

If the rover loses track of a target it either executes an action to re-acquire that target, or else foregoes any further use of that target.

If the rover uses significantly different time or energy during execution from what was predicted, it will go to other locations or attempt other measurements, more consistent with remaining time and energy, yet also scientifically valuable.

If the rover encounters a mission endangering fault condition (excessively low battery), it will immediately take remedial action to resolve the situation and continue with the plan, or else safely wait for the next communications opportunity.

[If the rover detects something of scientific interest, such as layered rocks, it will follow up with additional measurements, if allowed by remaining resources and tasks still to be done.]

Rover automatically, and safely, acquires contact measurements from desired locations on targets once it reaches them. The rover ensures that it can safely acquire the requested measurement, since the originally chosen target point might not have been properly tracked, or there is a hazard that was not apparent to mission controllers. If the rover cannot safely acquire a contact measurement it attempts to find a safe place as close as possible to the desired point on the same target rock.

6) Rover takes image panorama of search site (10m radius around rover). This will be used for target selection on the next sol.
7) Data products (excluding panorama) are down-linked to mission control, processed, and immediately made available to mission scientists.
8) Data Panorama down-linked overnight to mission control, to be available at start of next days target designation meeting.
9) Goto 1 for next sol

General Assumptions/Comments/Restrictions

- Short duration (Up to 30 minutes) contact measurements are sufficient for science goals. The impact of this technology is lessened as the rover needs to spend longer times at each target.

  This assumption can be relaxed if long duration measurements are done at end of the day, after the rapid investigation of many other targets, or if planning window extended to several days.

- Only one command cycle is allowed per simulated sol (activity planning, upload, sequence execution, data product download). Relaxing this trivially increases productivity and robustness. In practice (MER) there is some ongoing traffic between the vehicle and mission control throughout the sol in order to verify vehicle activities and get status updates.
- We use MER like rover, but featuring more computation power (modern laptop). Consistent with compressed demo timeline, static Mars environment and likely future availability.
- MER like (Gusev) terrain. Complexity is ultimately limited by capability of available off the shelf navigation and motion planning to be used onboard rover.
- Communications:
    - 1 uplink/downlink cycle per simulated sol (at beginning and end of day)
    - Downlink new panorama at end of simulated sol + all data products.

We are not implementing the ground based map merging necessary to relax this requirement.

Relevance to the Mars Exploration Program

In the March 2004 version of the Mars Exploration Program Analysis Group (MEPAG) document prepared by Taylor *et al*., as well as in all its past iterations, the investigation of ample samples and ground-truthing of orbital data through rock and soil sampling both appear as absolute and critical necessities to increase our knowledge of Mars, making the investigation of the geological diversity of Mars and related technique development top priorities.

New techniques that propose to significantly increase the number of samples that could be documented during the lifetime of a mission, such as the one proposed in this IS demonstration, will have tremendous impact on science productivity at almost all levels for and for most of the goals listed at key for NASA in its Mars Exploration Program (see Tables 1A to 1C) where *in situ* investigation and/or acquisition of samples is central to the science.

Examples of such MEPAG goals and investigations heavily relying on science target (e.g., rocks) investigation efficiency are provided in the tables below.

TABLE 1A: RELEVANCE TO MEPAG GOALS AND INVESTIGATION – GOAL 1(*)

GOAL 1: DETERMINE IF LIFE EVER AROSE ON MARS: Understand the habitability of Mars		
Objective A: Assess the past and present habitability of Mars (global and local scale, MEPAG 2004, p.5)	***Investigations 1****: Establish the Current Distribution of Water in all its Forms on Mars*	"To understand the conditions that gave rise to these potential habitats by characterizing their geologic and climatic context. (e.g., hydreous minerals)"
	Investigation 2*: Geological History of Water on Mars*	"…Thorough investigation of geological deposits that have been affected by hydrological processes."
Objective B: Characterize Carbon Cycling in its Geochemical Context	***Investigation 2****: Characterize the distribution and composition of inorganic carbon reservoirs on Mars through time*	"….Search for carbonate minerals from orbit, *in situ*…"
Objective C: Assess whether life is or was present on Mars	***Investigation 3****: Characterize the morphology or morphological distribution of mineralogical signatures*	"…Example measurements may include micron to nanometer imaging and chemical analysis of crystals or morphological characterization of sedimentary laminations."

** Based on the March 15, 2004 MEPAG document produced by Taylor et al.,*

TABLE 1B: RELEVANCE TO MEPAG GOALS AND INVESTIGATION- GOAL 2(*)

GOAL 2: UNDERSTANDING THE PROCESSES AND HISTORY OF CLIMATE ON MARS: Understand how the climate of Mars has evolved over time to reach its current state, and what processes have operated to produce this evolution		
Objective B: Characterize Mars' Ancient Climate Processes Through Study of the Geologic and Volatile Record of Climate Change	***Investigation 4****: Find physical and chemical records of past climates*	"…This investigation centers on finding geomorphic and chemical evidence of past climates…It requires determining sedimentary stratigraphy and the distribution of aqueous weathering products."
	Investigation 5*:Characterize the stratigraphic Record of Cliimate Change…*	" Studying the layered deposits as a key to understanding the climatic and geologic record…"

** Based on the March 15, 2004 MEPAG document produced by Taylor et al.,*

TABLE 1C: RELEVANCE TO MEPAG GOALS AND INVESTIGATION – GOAL 3(*)

GOAL 3: DETERMINE THE EVOLUTION OF THE SURFACE AND INTERIOR OF MARS: Understanding the composition, structure, and history of Mars is fundamental to understanding the Solar System as a Whole, as well as providing insight into the history and processes of our own planet		
Objective A: Determine the Nature and Evolution of the Geologic Processes that Have Created and Modified the Martian Crust	***Investigation 2****: Evaluate fluvial, subaqueous, and subaerial sedimentary processes and their evolution through time*	"…Understanding sedimentary processes requires knowledge of the age, sequence, lithology, and composition of sedimentary rocks…environmental conditions, mechanics of weathering, cementation and tranport processes."
	Investigation 6*: Determine the large scale vertical structure, chemical and mineralogical composition of the crust and its regional variations…*	"…Determining these structures requires global and local remote sensing, detailed geological mapping, and determination of mineralogy and composition of surface material."
	Investigation 9*: Determine the processes of regolith formation and subsequent modification, including weathering and diagenetic processes*	"…Understanding regolith formation and modification requires quantitative measurements of mineralogy, chemistry, and physical parameters of the surface and shallow subsurface

** Based on the March 15, 2004 MEPAG document produced by Taylor et al.,*

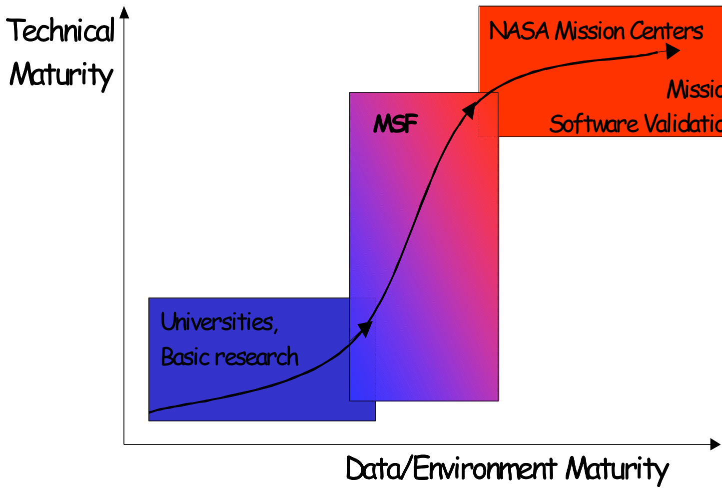
In addition to the goals and investigations specifically mentioned in Table 1A-1C, a future and critical objective of the Mars Exploration Program is to bring back a sample from Mars (Mars Sample Return Mission –MSR). The selection of a few grams of samples on Mars will have to be made extremely carefully for the scientific pay-off to be high relative to the cost of bringing the sample back to Earth. It will only come after the science community has obtained a better knowledge of Mars geological –and possibly biological environment. More effective sampling techniques will speed up that process and provide a clearer understanding of this environment, hence a better chance to select a critical sample to return.

This IS demo is providing the very first step in a direction that will make the remote robotic science operations on Mars more efficient by initiating the quantification of such parameters. It will help increase the speed at which we understand the Martian environment, provide MEP and MEPAG with methods facilitating successful achievement of their goals and objectives, and in the long- term, lowering mission cost by increasing mission productivity.

In order to achieve a sustainable presence in space, NASA will rely heavily upon autonomous and semi-autonomous robots collaborating with human astronauts. Simulation-based technology development is a powerful catalyst for furthering space exploration by accelerating the development of critical technology. In a simulation environment, model validation catches errors earlier in the design process avoiding costly redesigns and increasing reliability. Simulation-based development uses commodity level computing hardware that is cheaper and more reliable than the expensive purpose-built hardware platforms normally used for robotic software development. With a simulation-based approach to technology advancement, particularly the definition and evaluation of robotic software, the resulting capabilities will be advanced, cost-effective, and robust.

Planetary surface exploration missions require technology that is extremely reliable and predictable. The migration path from laboratory to mission is a difficult one in terms of technology, politics, and funding. The MSF hopes to offer a steppingstone toward mission readiness for control software that would otherwise remain unproven.

The graph below illustrates conceptually a migration path for planetary missions and the role of the MSF in maturing technology at NASA. Providing the MSF architecture as open source to universities allows the focus of research to be on autonomy, rather than developing a testing environment. The algorithms and technology developed using the MSF will be more easily transferred for evaluation at NASA. Because the MSF integrates and provides access to technology standards such as ROAMS and CLARAty familiar to Mission managers, the resulting algorithms will be more easily integrated and validated for use in future missions.

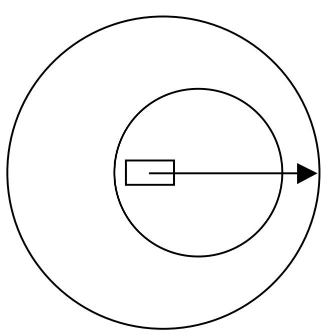


MSF role in technology migration.

Performance Goals

Demo Floor: The minimum performance level required so as not to impact other components of integrated demonstration.

Extended: Desired competencies to be accomplished subsequent to this demonstration,

Metrics	**Demo Floor**	**Baseline**	**Extended**
Number of science targets investigated per command cycle	2	3-5	10
Size of investigation site 	All targets within 5m of each other and 2.5m of rover start position.	All targets within 10m of each other and of rover start position.	All targets within 20m of each other and of rover start position.
Final instrument placement precision, with respect to point chosen by users.	On same rock	Within 5cm, precision roughly characterized	Within 1cm, precision well characterized.

Rover Resource and time constraints enforced	Battery level > T (user defined)	Execution time < 3hrs (user defined) Time windows for measurements (user defined)	
Discrete Faults	Reduced tracking precision ➔ place anywhere on rock Arm collision ➔ abort measurements.	Unrecoverable loss of target ➔ don't go there. no safe placement ➔ place on closest safe area of rock.	Unrecoverable loss of target ➔ re-evaluate plan to optimally exclude further measurements from that target.
Follow-up measurements for science opportunities not known to mission control	None	Mission control specifies where to look. Find naturally occurring layered rocks Respond with science cameras (0 resource)	Systematic, background process to continuously search area around rover for layered rocks. Plan re-evaluation to permit additional measurements requiring non-zero resources.
Ground Data Systems			
Simulation-based Testbed Environment	Representative virtual Mars terrain	Virtual version of Marscape test site	Virtual version of Marscape test site with additional rock placement
Simulation-based Testbed	Interface to	Interface to	Interface to

Interfaces	virtual vehicle executes a robust set of commands for software testing.	virtual vehicle accommodates all functionality required of real hardware test platform.	virtual vehicle identical to real hardware test platform.
Simulation-based Test Scenarios	Test scenarios in simulation include Mars terrain library, conditional execution, and measurements of power and time usage.	Test scenarios in simulation include virtual rendition of Marscape test site, realistic power and time models, uncertainty in measurements and consumables, and measurements of power and time usage	Test scenarios in simulation include configurable virtual environment, realistic power and time models, uncertainty in measurements and consumables, target tracking, and user capability to inject failures.

Note: MEPAG and the Astrobiology Roadmap do NOT specify any minimal performance targets specifically for these metrics. Cannot specify *a priori* required performance numbers for all metrics

Technology Components

Autonomous Navigation and Instrument Placement

Goals

Autonomously navigate to multiple (3 or more) science targets up to 10m distant and place an arm mounted instrument (microscopic camera) on a target rock, as close as possible to the target location chosen by users, yet consistent with arm and instrument safety requirements.

This requires that

- Rover visually keeps track of multiple (3-5) science targets (rocks), up to 10m distant from rover, with sufficient precision for an instrument placement.
- Rover keeps track of locations related to science targets, such as points where it must be in order to make a measurement on a given target.

- Rover navigates to specified points designated with respect to a science target with sufficient precision
- Target hand-off from mast mounted cameras (used for tracking) to front mounted hazard cameras (used for instrument placement)
- Rover recognize target rock and confirm that target point is consistent with instrument constraints.

Technical Approach

Instrument Placement

The first step in determining where to place an instrument anywhere on a rock target (or other large area) is to obtain a 3D scan of the work area. This can be done with stereo cameras. It is important that they be well calibrated with respect to the rover manipulator arm, as the derived 3D point cloud will be used to compute desired instrument poses.

Next, the rock (or target area) in the 3D model of the work area must be segmented from the background. We have developed an iterative 3D clustering algorithm, based on the statistical EM algorithm, for this purpose. This algorithm is very robust to noise, requiring only that the ground be relatively flat (but at an arbitrary orientation) and the work area have at most one rock significantly larger than any clutter in the scene.

Next, all points on the target rock, within the arm workspace, are checked for consistency with the rover instrument to be placed:

i) Confirm that requested target point is safe

ii) Find safe point on rock closest to requested point OR

iii) Find safest places on rock

A placement position on the rock is considered safe if:

- All points on the rock within a given radius do not deviate more than a preset value from the best fit plane.
- There exists a collision free path to place the instrument there.

Next it goes to a pose near the highest priority target pose in the workspace, holding back a safe distance along the target surface normal. To compensate for possible small errors in surface location, the instrument's final approach is along the measured normal to the target rock face, moving slowly forward until contact is confirmed by mechanical sensors.

Navigation to Targets (SIFT based Keypoint Tracker)

To be written.

Mesh Registration for Hand-off

For every pixel in the left camera image for which a correspondence is found in the right camera image, our stereo algorithm estimates the depth to that point. These depth estimates are combined to produce a 3D model of the surface. If two models of a surface are made from different locations, the rigid transformation that aligns the two models can be used to determine the coordinate transformation between views.

The surface models are represented by triangulated meshes with vertices **v** and **v'**. If the two 3D models contain some region of overlap, there is a rigid transformation that aligns the overlapping regions. We represent the rigid transformation using the parameter vector $\mathbf{p} = (x,y,z,\alpha,\beta,\gamma)^T$ corresponding to 3 translational and 3 rotational degrees of freedom. These parameters define a transformation matrix $\mathbf{T_p}$. If **p** is the parameter describing the transformation between surfaces **v** and **v',** then for every pair of corresponding points $\mathbf{v_i}$ and $\mathbf{v_i}'$, the relationship

$$v_i' = T_p v_i$$

holds. With real observations this equality will not hold exactly.

Our mesh registration approach projects these two models into a virtual range sensor view and minimizes the difference between the rendered depths at each point. The rendering takes *O(n)* operations, where *n* is the number of pixels in the virtual range sensor. For each triangle on the mesh **v'**, the vertices $\mathbf{v_i}'$, $\mathbf{v_j}'$, and $\mathbf{v_k}'$ are projected onto the image plane. For every pixel inside that triangle, the location of the intersection of the camera ray $\mathbf{c_z}$ and the facet of the mesh is a point $\mathbf{s_i}'$, given by

$$s_i' = \alpha_i v_i' + \alpha_j v_j' + \alpha_k v_k'$$

with $\alpha_i+\alpha_j+\alpha_k=1$. The depth to the intersection point is the z coordinate in the camera frame,

$$z_i = \mathbf{c_z} \cdot \mathbf{s_i}'$$

The vector of all depths z_i is denoted **z**. The surface model **v'** does not move during registration, so **z** is a constant.

The depth to point $\mathbf{v_i}$ changes with transformation **p**.

$$\mathbf{s_i} = \mathbf{T_p}(\alpha_i \mathbf{v_i} + \alpha_j \mathbf{v_j} + \alpha_k \mathbf{v_k})$$

$$h_i(\mathbf{p}) = \mathbf{n_c} \cdot \mathbf{s_i}$$

We define a robust objective function which is the sum of the absolute deviations between the projected depths:

$$J(\mathbf{p}) = \sum | h_i(\mathbf{p}) - z_i |$$

Because the J(**p**) has local minima, we first perform a coarse correlation search in order to narrow down the location of the best solution. Our initial estimate of **p**, $\mathbf{p_0}$ comes from the stereo SIFT-based tracker described above. Consider that **p** is decomposed into a rotational component **r** and a translational component **t.** Furthermore, consider that **t** is decomposed into:

$$\mathbf{t} = x\mathbf{c_x}, + y\mathbf{c_y} + z\mathbf{c_z}$$

where $\mathbf{c_x}$, and $\mathbf{c_y}$ are in the plane of the virtual range sensor, and $\mathbf{c_z}$ is the pointing direction of the sensor. Because a search over the 6 dimensions of **p** is expensive, we make the following approximations:

- For small changes in **t**, $h_i(x,y,z+ \Delta z,\mathbf{r}) \sim= h_i(x,y,z,\mathbf{r}) + \Delta z$. In other words, a change in transformation along the z axis of the virtual range sensor by some distance Δz changes h_i by approximately the same amount.

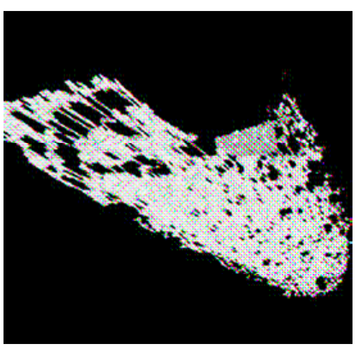
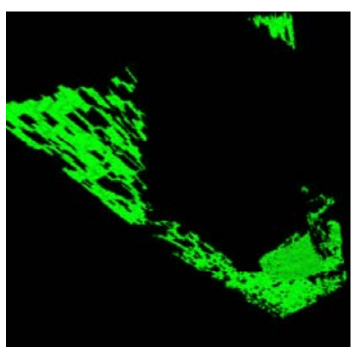
- Our initial estimate of **r** is approximately correct

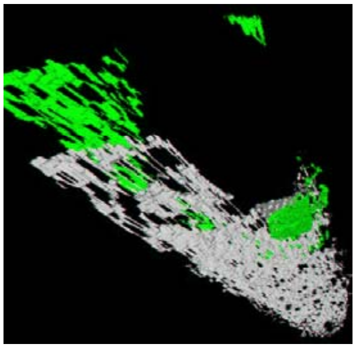
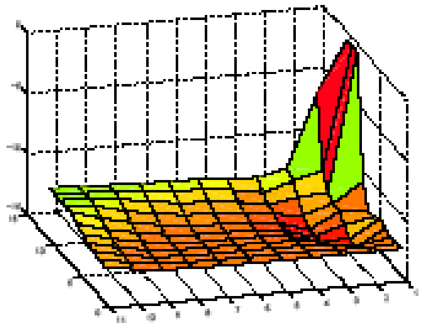
These two approximations allow us to perform the correlation search across only two dimensions: the the x and y axis of the virtual range sensor.

For every Δx and Δy searched, the transformation **p** is computed by translating initial estimate $\mathbf{p_0}$ by Δx and Δy by translating in the directions of the x and y axis of the virtual range sensor. The correction Δz to z_0 which minimizes the objective function $J(x_0+\Delta x, y_0+\Delta y, z_0+\Delta z, \mathbf{r_0})$ is calculated as follows:

$$\Delta z = \mathit{median}(h_i(x_0+\Delta x, y_0+\Delta y, z_0, \mathbf{r_0}))$$

At the time of testing contained in this report, the results of the coarse correlation search are being used directly.

Assumptions

Assumptions

- Off the shelf rover arm path planning
- Off the shelf inverse kinematics
- Off the shelf arm control. We do not address precision manipulation tasks.
- MER level obstacle avoidance (GESTALT lite) available on K9
- Deduced reckoning available on K9 with error growth approximately 10% of distance traveled.
- New image panorama, taken from start rover position, available for target templates, and that rover does not move in between panorama acquisition and users uploading new target templates.

Demonstrations and Results

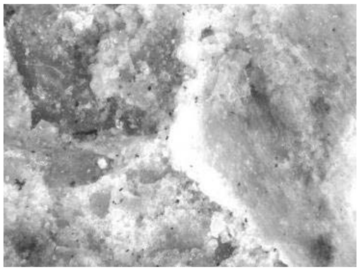
<table>
<tr><td>Date : August 2002</td><td>Location: NASA Ames Research Center Marscape test facility</td></tr>
<tr><td colspan="2">Goals: demonstrate autonomous instrument placement anywhere on a rock in front of rover

Results: <u>Videos\instrument placement\instrument placement 2001.mov</u></td></tr>
</table>

Autonomous instrument was successfully demonstrated on the K9 rover. It approached a target from a distance of 2m, driving forward in a straight line using odometry and deduced reckoning. The outdoor test site had moderate clutter, including scattered cobble and loose soil. The target rock itself is a complex aggregate of two rocks, one with a smooth surface and the other one grossly misshapen. There was a variety of textures and colors in the scene.

Using images from it's front stereo hazard cameras, K9 autonomously assessed to rock scene: segmenting the rocks from the ground and deciding on the optimal place, in its workspace, to place an arm-mounted microscopic camera.

Once the arm placed the camera, we obtained microscopic images of the rock surface.



This demonstrated single cycle instrument deployment from a Mars rover in an outdoor test environment of intermediate complexity.

Successful instrument placement on most rocks, demonstrated in Ames Marscape and in quarry location:

Date : October 2003	**Location:** GraniteRock Aromas quarry, Watsonville, CA and NASA Ames Research Center
Goals: Integrated end-to-end demonstration of rover tracking 2 targets using mesh registration for target tracking, followed by instrument placement against 1 of the targets. **Results:** Videos\ERT1-2003.mov	

Goals accomplished. See integrated demonstration section or movie

Date : August - July 2004	**Location:** NASA Ames Research Center Marscape test facility
Goals: Test key-point tracker using wide field of view navigation cameras and narrow field of view science cameras **Results:** see below	

Navigation Cameras:

Videos\tracking\navcam-tracking 07-29-04.mov

Note recovery from momentary occlusions of target. At least one cycle of complete occlusion is tolerated:

Science Cameras:

Target tracking using narrow field of view rover science cameras, with very precise results:

Videos\tracking\scicam-tracking 8-31-04.mov

Tracking with science cameras proved ultimately more robust because of greater texture detail and stereo accuracy (yielding more and better keypoints).

Other trackers require very frequent updates or accurate inertial navigation to function with images from science cameras, which would otherwise have insufficient overlap between pairs.

Date : September 20, 2004	**Location:** NASA Ames Research Center Marscape test facility
Goals: demonstrate autonomous instrument placement on safe location as close as possible to target point on a rock **Results:** Videos\instrument placement\ptzmovie24fps.mov Videos\instrument placement\ipviz.mov	

Goals achieved:

Date : September 23, 2004	**Location:** NASA Ames Research Center Marscape test facility
Goals: demonstrate tracking of, navigation to and instrument placement on 3 consecutive targets at different distances, up to 10m.	

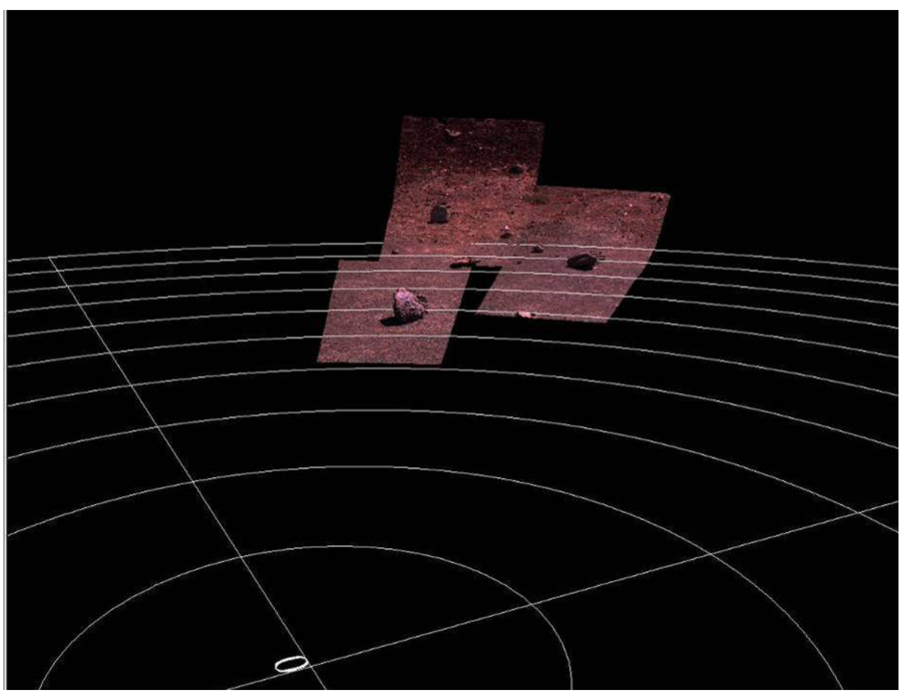


Figure 1 : 9/22/2004 Target arrangement

Successful tracking and consecutive navigation to each target, in order. Succesful location of safe locations for instrument placement, but actual placement not possible due to hardware failure in manipulator arm.

Table 2 9/22/2004 Performance

	Target 1 (5m)	Target 2 (7.5m)	Target 3(10m)
Time to reach target	21 minutes	+ 42 minutes	+17 minutes
Tracker accuracy (keypoint tracker)	0.68 cm	0.29 cm	1.3 cm
Hand-off accuracy (3D mesh registration)	5.1cm	2.8cm	3.9cm
Final placement accuracy	n/a	n/a	n/a

Target 1	Target 2	Target 3
		
		

Figure 2 Keypoint tracker images

Date : September 23, 2004	**Location:** NASA Ames Research Center Marscape test facility
Goals: demonstrate tracking of, navigation to and instrument placement on 3 consecutive targets at different distances, up to 10m.	

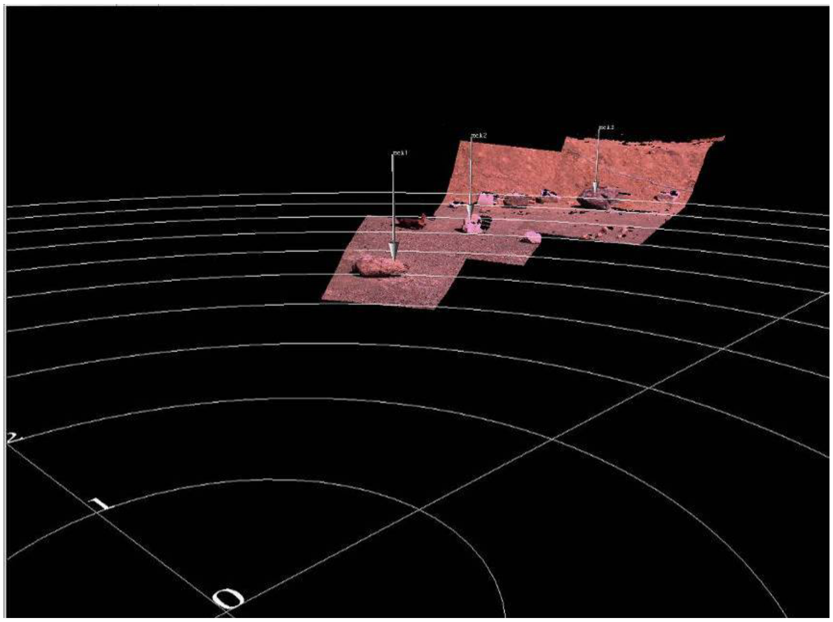


Figure 3: 9/23/2004 Setup

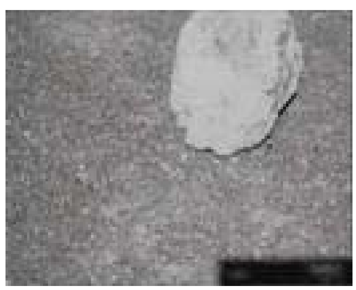
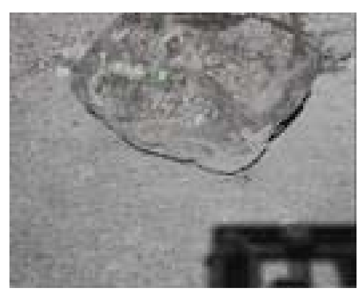
Successful tracking and consecutive navigation to, and instrument placement on each target, in order shown.

Table 3: 9/23/2004 Performance

	Target 1 (5m)	Target 2 (7.5m)	Target 3(10m)
Time to reach target	25 minutes	+ 27 minutes	+23 minutes
Tracker accuracy (keypoint tracker)	~ 0.3 – 2.3 cm	Tracker failed due to exceptional lighting contrast on target preventing acquisition of good stereo models. System partially recovers using odometry and mesh registration (below)	1.7 cm
Hand-off accuracy (3D mesh	3.5 cm	~20cm	4.2 cm

registration)			
Final placement accuracy*	~6.3 cm	~11 cm	~ 3 cm

* Placement accuracy varies depending on rock surface. System attempts to place at safe point closest to target. If target not safe, system will not place on it (hence accuracy less than hand-off).

Target 1	Target 2	Target 3
		
		
		
		

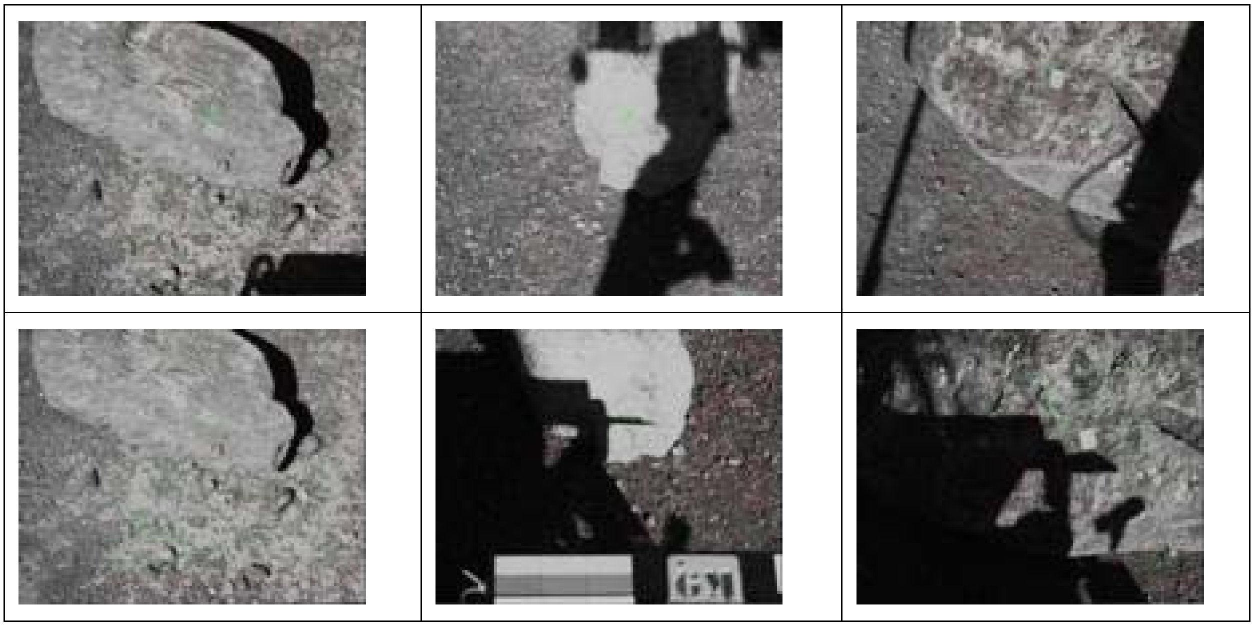


Figure 4: Keypoint tracker images

Performance

K9 has demonstrated to navigation to 3 consecutive targets, up to 10m distant, and has placed the CHAMP microscopic instrument within 5 cm of designated points (when safe).

All performance goals have been met, as indicated above.

Contingency Planning and Robust Execution

Goals

In order to accomplish the task of instrument placement within a single cycle with the robustness required for a mission, the on-board software must be able to handle failures and uncertainties encountered during the component tasks. A task may fail, requiring recovery or retrying. Tasks may exhibit a high degree of variability in their resource usage, using more (or less) time and energy than expected. Finally, the state of the world and the rover itself is only predictable to a limited extent. Exploring multiple rock targets further exacerbates this situation. These factors require that the rover's software have the ability to reason about a wide range of possible situations and behaviors. A simple script is insufficient; instead, the rover can use either on-board task planning or off-board planning in conjunction with robust on-board execution.

Off board generation of flexible plans for rover to acquire measurements of greatest value from multiple targets subject to time and resource constraints and the large expected uncertainty in rover performance due to the complexity and duration of the anticipated task.

- Resources constraints:

- Battery level always above threshold (Tf) at end of day (user specified), and at all time (Ta)
- Total duration of days activities > T_time (user specified)

- Time constraints:
  - Certain measurements only permissible between local time window (user specified)
  - Execution time
- Faults:
  - Loss of target tracking
  - Inability to acquire measurement

Technical Approach

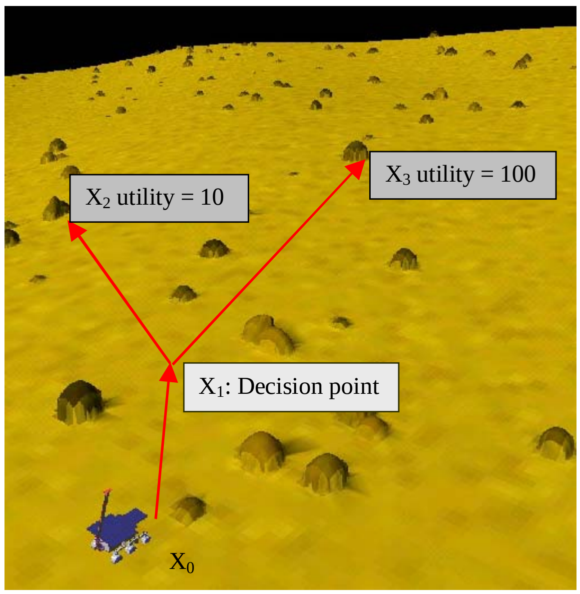


Figure 5 Waypoint and utility planning for instrument placement.

The contingent planning system determines which of the objectives to pursue along with the detailed commands necessary to achieve those objectives. In addition, it also inserts "contingency branches" into the plan to cover situations where the plan might possibly fail. In the example shown in Figure 5, suppose the planner initially constructs a plan to go to waypoint X_1, and then location X_3. It could then add a contingency branch to go to X_2 instead, if, upon arrival at X_1, there is not enough power or time available to continue to X_3.

PICo: Planning Incrementally with Contingencies.

Given a set of objectives and their associated values, the PICo planning system determines which of the objectives to pursue along with the detailed commands necessary to achieve those objectives. In addition, it also inserts "contingency branches" into the plan to cover situations where the plan might possibly fail. This contingency planning is done using an incremental Just-In-Case approach [5], as illustrated in Figure 6. First a “seed” plan is generated having maximum expected utility. That is, the plan achieves the best objectives possible given the expected resources available (time and energy), and expected consumption of those resources by the actions involved. This plan is then evaluated to determine where it might fail, given uncertainty in resource consumption by the various actions involved. A branch point is then chosen by using a heuristic that evaluates the potential to add branches at the end of every action in the seed plan and recommends the “best” branch point along with the branching condition and the goals to pursue. An alternative or *contingency* plan is then constructed for this branch, and incorporated into the primary plan. The resulting conditional plan is again evaluated, and additional branches can be added as needed.

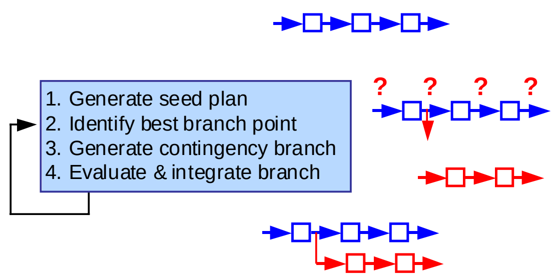


Figure 6 PICo algorithm

The contingency planner makes use of the PLASMA planning engine to generate seed plans, and to generate the plans for the contingency branches. For constructing a seed plan, the contingency planner gives PLASMA the goals, expected resource availability, and expected resource consumption of actions. When the plan comes back, the contingency planner evaluates it using a Monte Carlo simulation to determine the impact of uncertainty in resource usage and tracking failure. To build the branch, the planner passes appropriate goals, the state of the rover at the branch point, and resource availability to PLASMA. The state of the rover and the resource availability is based on the branch condition which describes the amount of resources (time and energy) or whether tracking of a target has failed.

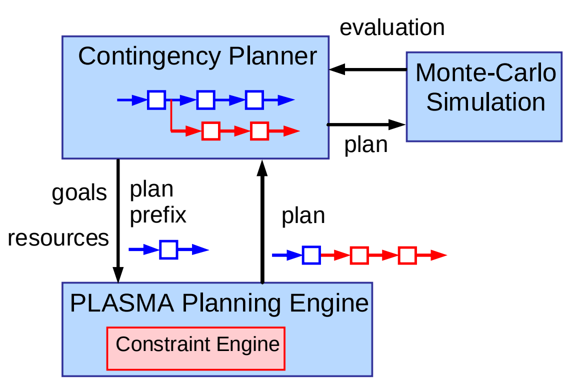


Figure 7 Architecture of the contingency planner.

The problem of automatically choosing good branch points and good branch conditions is quite hard in general (see **Error! Reference source not found.**, **Error! Reference**

source not found. for details). Intuitively, it might seem that a good place to put a contingency branch is at the place where the plan is most likely to fail. Unfortunately, this is often near the end of the plan, when resources (time and energy) are nearly exhausted. With few resources remaining, there may not be any useful alternative plans. Instead, one would like to anticipate failures earlier in the plan, when useful alternatives remain. In other words, the planner is looking for the point(s) in the plan where a contingent branch could be added that would maximally increase the overall utility of the plan. This quantity is very difficult to compute because it is a function of the resources which are uncertain continuous quantities.

Because it is very difficult to identify the best point to branch along with the branching condition and goals to pursue, we instead use heuristics. Among the many different heuristics we have implemented the most noteworthy is the OP heuristic.

This heuristics uses:

- A representation of the planning problem as a deterministic orienteering problem (OP) to evaluate what are the best trajectories to follow from a given situation;
- A Monte Carlo simulator to evaluate the resource distribution at each step of the plan.

At the first call, when the plan is empty, it uses the solution of the OP to recommend an initial set of goals. At subsequent calls, it walks through the whole plan and evaluates the benefit of adding each type of branch point at each possible place. This evaluation is performed by multiplying the utility of the best possible branch we could add (obtained

by solving the deterministic OP) by the probability of falling into a region of the state space where we would prefer to branch (given by the Monte Carlo simulator). It then adds the branch with he highest heuristic value.

Robust Execution

The CRL Executive is responsible for interpretation of the contingent plan coming from the ground and generated by the contingent planner. The CRL Executive is designed to be more capable than traditional sequence execution engines; it can handle the expressive plans generated by the contingent planner and can perform limited plan adaptation itself.

The planner translates its plan into the Contingent Rover Language (CRL) for uplink, and the CRL Executive interprets the CRL-encoded plan directly. CRL is a flexible, conditional sequence language that allows for execution uncertainty **Error! Reference source not found.**. CRL expresses temporal constraints and state constraints on the plan, but allows flexibility in the precise time that actions must be executed. Constraints include conditions that must hold before, during, and after actions are executed. A recent addition to CRL is the ability to specify concurrent threads of activity.

A primary feature of CRL is its support for contingent branches to handle potential problem points or opportunities in execution. The contingent branches and the flexible plan conditions allow a single plan to encode a large family of possible behaviors, thus providing responses to a wide range of situations.

The structure of the CRL plan language and its interpretation are completely domain-independent. Domain-dependent information is added by specifying a command dictionary, with command names and argument types, and a command interface, which passes commands to the rover and return values and state information from the rover.

The CRL Executive is responsible for interpreting the CRL command plan coming from ground control, checking run-time resource requirements and availability, monitoring plan execution, and potentially selecting alternative plan branches if the situation changes. At each branch point in the plan, there may be multiple eligible options; the option with the highest expected utility is chosen. For this demonstration, the contingent planner generated mutually exclusive branches.

A novel feature of the CRL Executive is its support for "floating contingencies," which are plan fragments that may be inserted at any point in execution **Error! Reference source not found.**. For example, a plan to perform opportunistic science during a traverse is naturally expressed as a floating contingency, since the presence and position of an interesting science target is unknown before the traverse. Likewise, a plan to stop and recharge the battery is another example of a floating contingency. In general, floating contingencies would be impractical for the planner to consider because of the large number of possible branch points that they would add to a plan.

The CRL Executive is implemented as a multi-threaded, event-based system (see Figure 8). Around a central Executive event-processing loop are threads to handle timing, event monitoring, action execution monitoring, and telemetry gathering. The central event processor sends requests to the other threads (for example, "wake up at time 20" or "notify when battery state of charge is below 4Ah") and receives events relevant to those requests. This architecture allows the CRL Executive to support concurrent activities and flexible action conditions expressible within the CRL language.

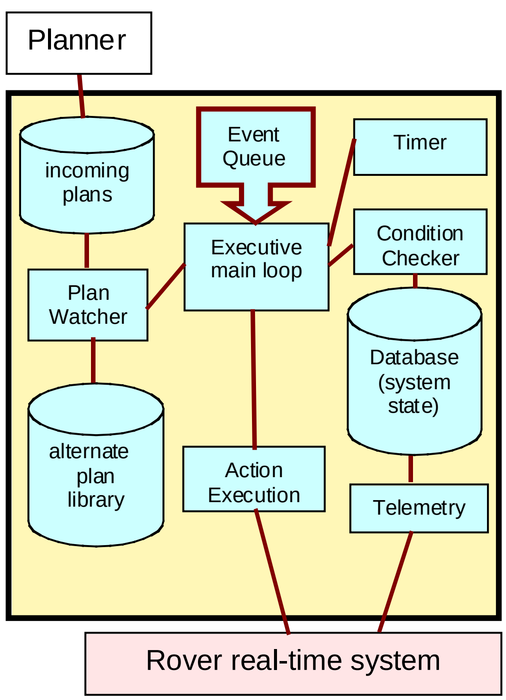


Figure 8 CRL Executive structure. The main event loop communicates with other threads for services such as timing, action monitoring, and event monitoring. External connections are to a planner, which supplies new plans to execute, and a rover real-time system, which executes actions and supplies telemetry data.

Assumptions

- Adequate time and power use models for each task
- Adequate battery level model (K9 will have simulated battery)

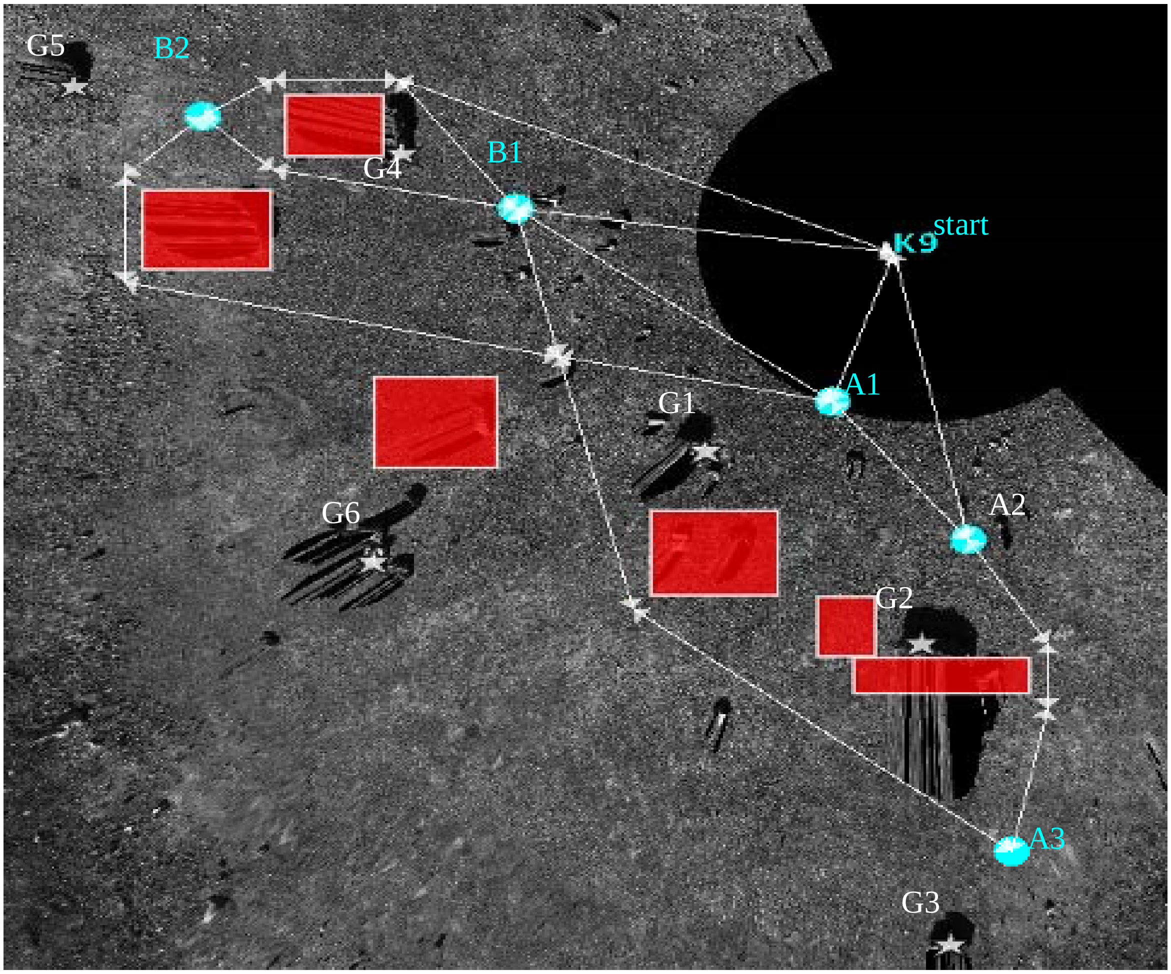
Results and Demonstrations

Date : October 2003	**Location:** GraniteRock Aromas quarry, Watsonville, CA and NASA Ames Research Center
Goals: Contingent plan generation, producing CRL plans that branch on energy, and execute on K9 rover	

Goals achieved. See Integrated Demonstrations section of document.

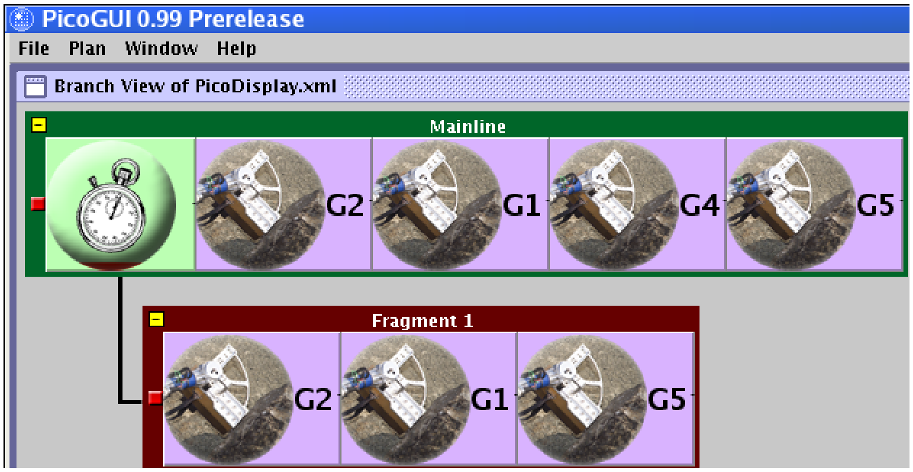
Date : September 23, 2004	**Location:** NASA Ames Research Center
Goals: Contingent plan generation, showing plans that branch on: - Time - Energy - Discrete failures (target tracking failure)	

The following mission problem is posed:



Observation Goal	Rover Location to achieve observation goal	Utility
G1 (CHAMP image)	A1	187
G2 (CHAMP image)	A2	150
G3 (CHAMP image)	A3	71
G4 (CHAMP image)	B1	111
G5 (CHAMP image)	B2	198
G6 (Science Camera Image)	A1	51

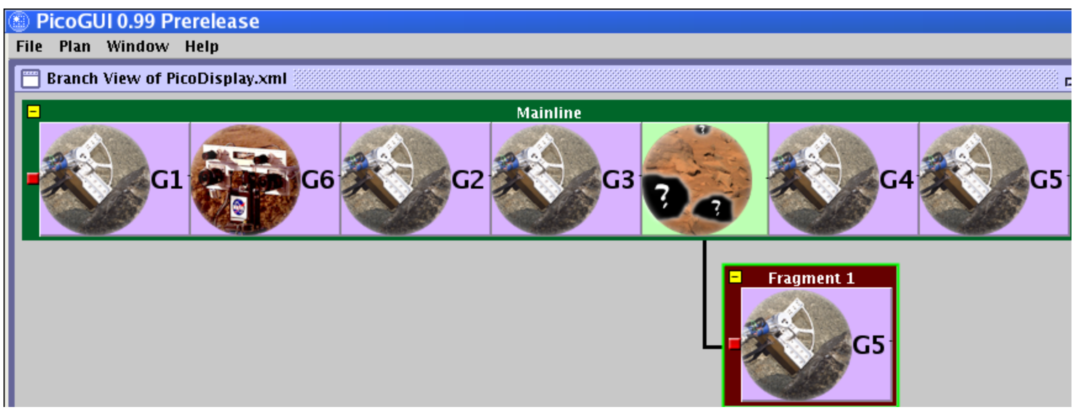
Case 1: Branch due to lack of time to complete the mainline



Mainline plan: Select goals in the decreasing order of the utility value/cost of achieving that goal. Goals selected in mainline are based on this criteria, given the cost of achieving each goal from the Start position.

Branch condition: If during the drive to G2 we are able to detect that we do not have sufficient time to complete all the goals in the plan, then branch and do the goals that have the sufficient value and can be done within the time left. This results in the dropping of goal G4. Even though G5 is further away than G4 and would take more time to achieve, it has more value and hence it is retained.

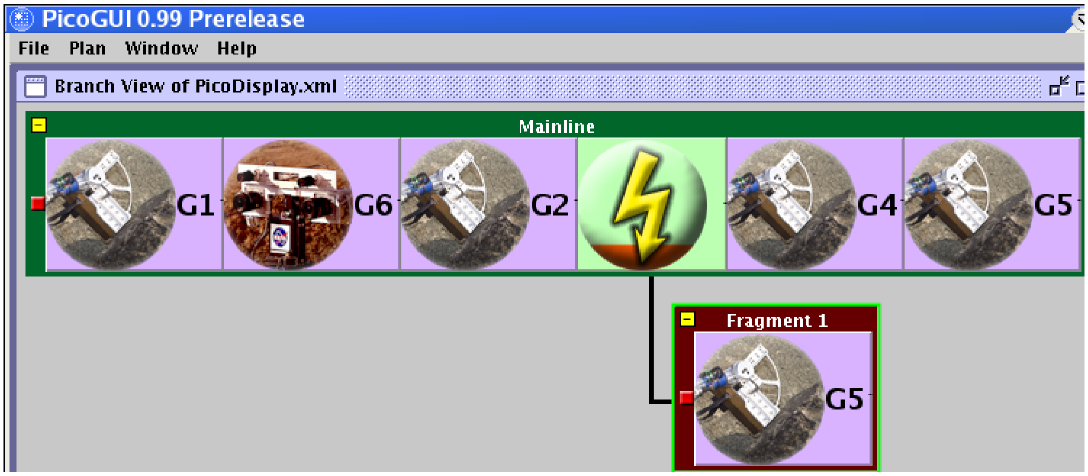
Case 2: Branching due to failure of tracking



Mainline plan: Select goals in the decreasing order of the utility value/cost of achieving that goal. Goals selected in mainline are based on this criteria, given the cost of reaching each target from the Start position.

Branch condition: There is a high risk of losing track of target RockB1 along the drive from RockA3 to RockB1 (via waypoints W7, C13). Branch condition specifies that if we lose track of RockB1, then we achieve only G5 (CHAMP@RockB2) because G4 (CHAMP@RockB1) is no longer achievable.

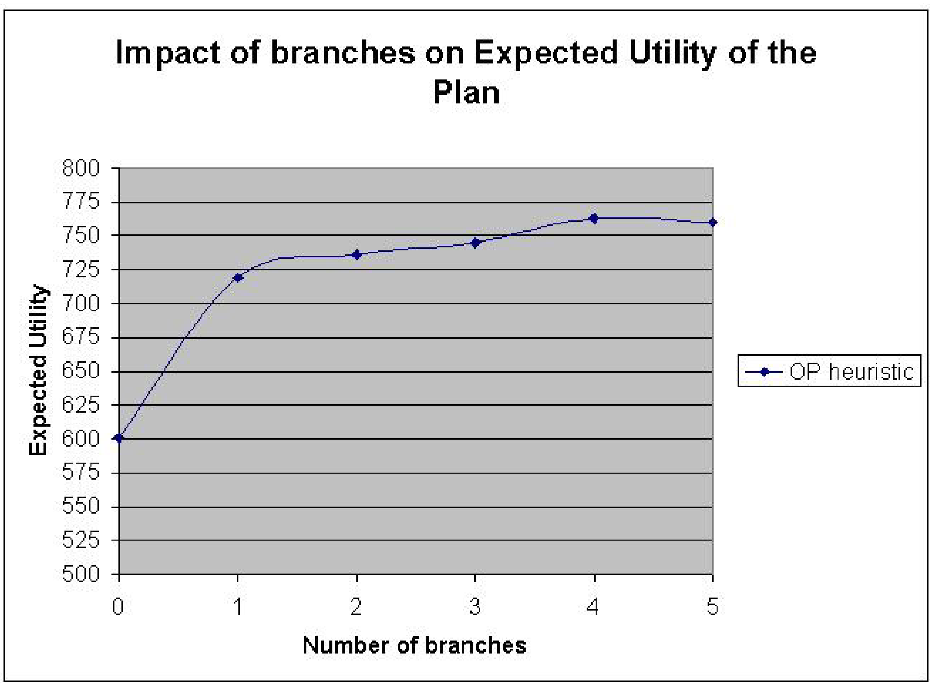
Case 3: Branching on lack of energy to complete the mainline.



Mainline plan: Select goals in the decreasing order of the utility value/cost of achieving that goal. Goals selected in mainline are based on this criteria, given the cost of reaching each target from the Start position.

Branch condition: After completing G2 (CHAMP@ RockA2) the rover does a long traverse to the other side of the rock field to reach Rocks B1 and B2. During this traverse, if the rover consumes more energy than expected such that it is no longer able to complete the mainline (both G4 & G5), then it drops goal G4. Again, G5 is chosen to be retained rather than G4 because G5 has more utility.

Performance



As we can see and expect, the expected utility value tends to increase with increasing number of branches, tending towards an optimal plan value threshold for the given planning problem. The heuristic employed for this scenario uses:

- a representation of the planning problem as a deterministic orienteering problem to evaluate what are the best trajectories to follow from a given situation;
- a Monte Carlo simulator to evaluate the resource distribution at each step of the plan.

At the first call, when the plan is empty, it uses the solution of the OP to recommend an initial set of goals. At subsequent calls, it walks through the whole plan and evaluate the benefit of adding each type of branch point at each possible place. This evaluation is performed by multiplying the utility of the best possible branch we could add (obtained

by solving the deterministic OP) by the probability of falling into a region of the state space where we would prefer to branch (given by the Monte Carlo simulator). It then adds the branch with the highest heuristic value.

References

Ground Data Systems

Goals:

1) Preliminary Site Analysis
2) Daily Mission Specification
    a. Efficiently designate multiple desired science targets and associated observations:
    b. Interact with planner to generate and visualize daily activity plan, and refine as necessary
3) Situational Awareness
    a. Organize and display returned data products from multiple targets in an intuitive way. Round trip data product tracking, so that returned products are correctly associated with requested observations.
    b. Display rover execution trace for users to ascertain what rover actually did.

Technical Approach

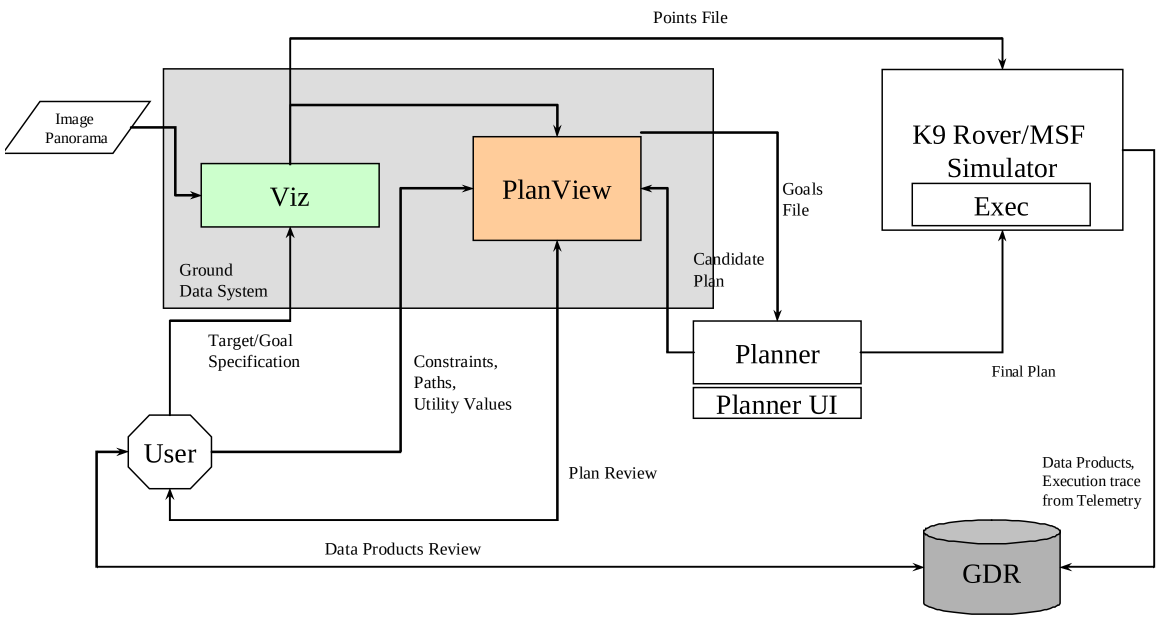


Figure 9: Ground Data System components (Viz, PlanView, GDR) and information flow

The Ground Data System is composed of 3 main components:

- **Viz**: a 3D photo-realistic immerse display program for visualizing 3D terrain models of the area around the rover, generated from rover stereo camera panoramas. Viz was originally developed for Mars Pathfinder, and used successfully on MER for a variety of geo-morphological measurements and virtual exploration of the area surrounding the rovers. Our version of Viz has been significantly enhanced to allow users to specify many science targets and observations.

- **PlanView:** large format, touch sensitive 2D user-interface, built on top of MERBoard/XBoard (developed for MER), for users to review requested observations and targets, specify additional daily mission constraints (including obstacles, time-of-day, and allowed paths), visualize plans and execution traces returned from the planner and rover respectively, and access returned data products.

Figure 10 : Mission Operations Center, with 3D Viz display (right) and PlanView (left).

- **Ground Data Repository:** data base and file system for organizing and storing all data related to a given mission day, including rover image panoramas, user requests, daily activity sequences and returned data products.

The user base interacting with GDS is made of 2 types of users:

Science team: Focused on the science goals associated with the rover operations

Rover Engineering team: Focused on the rover operations.

While the 2 user types will have differing requirements on the GDS, the two roles interact closely in the planning & execution of all tasks around the rover operations. The GDS architecture reflects this to enable clean and seamless coordination between the 2 users, who can both work together in the same room and use the same tools.

Task Breakdown

The following are the tasks that users must accomplish with the GDS:

1) **Preliminary Site Analysis –** users virtually explore the environment around the rover to decide what targets are interesting.
2) **Target Requests** – designate and name points of scientific interest in the environment, and extract information required for the rover to track them as it moves around:
3) **Observation requests –** designate observations to be acquired by the rover. An observation consists of:

a. Instrument Specification – which instrument (microscopic camera or science cameras) to use, and any required instrument parameters (exposure, focus positions, etc).

b. Target Points

c. Observation Points – where the rover must be in order to acquire the requested measurements.

d. Observation Utility – subjective measure of measurement value, used by planning system to prioritize measurements.

e. Observation Constraints – Time-of-day and precedence.

4) **Path Specification** – determine network of acceptable paths the rover may between the start position and all observation points. Acceptable paths are constrained by obstacles and the rover target tracking system. In order to specify paths, these must also be specified:

 a. Obstacles – areas, potentially including target rocks, where the rover should not go.

 b. Tracking Regions – areas within the rover must remain in order to track a particular target point.

5) **Activity Planning**– send observation and path data to the planner, visualize and refine the returned plan

Tasks 1-5 may occur concurrently, and through several iterations, until users are satisfied with the daily activity plan, which is modified by adding, deleting or changing observations, changing observation utilities and constraints requested observations and path network.

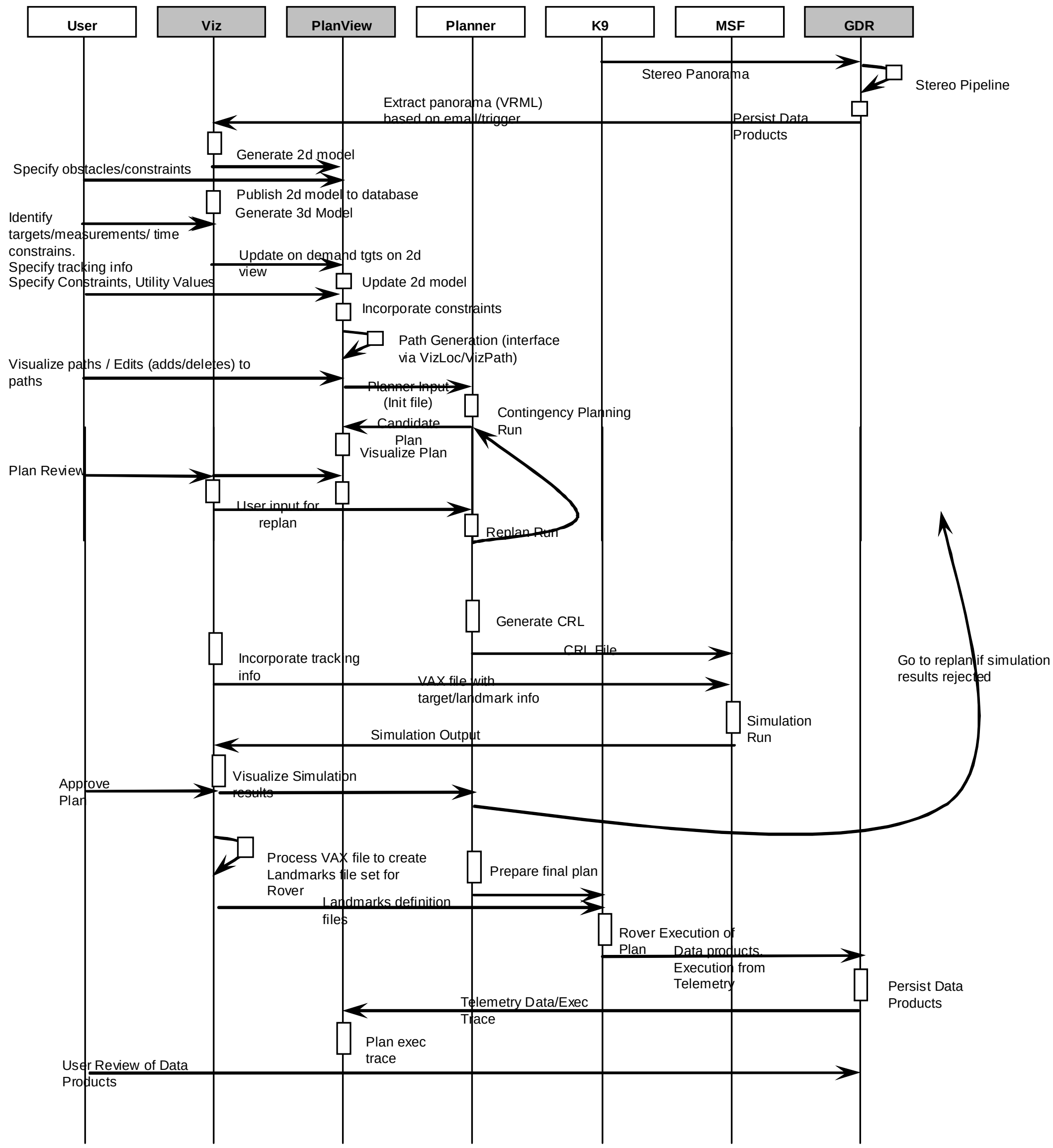
Once users are satisfied with the daily activity plan, it can be sent to the MSF simulator for a final sanity check, and then uploaded to the rover which executes it.

6) **Data Products Review** – upon sequence execution the rover returns significant data, including requested observation data produces and telemetry. Because of uncertainty, it is unlikely that the rover will have behaved exactly as predicted.

 a. Execution Trace – show the sequence of activities the rover actually accomplished.

 b. Data products – show which of the requested observations the rover acquired, and present them to the users for verification and analysis.

The following swim-net indicates the information flow corresponding to these activities, and which components do what:


User
Viz
PlanView
Planner
K9
MSF
GDR
Stereo Panorama
Stereo Pipeline
Extract panorama (VRML) based on email/trigger
Persist Data Products
Generate 2d model
Specify obstacles/constraints
Publish 2d model to database
Generate 3d Model
Identify targets/measurements/ time constrains.
Specify tracking info
Specify Constraints, Utility Values
Update on demand tgts on 2d view
Update 2d model
Incorporate constraints
Path Generation (interface via VizLoc/VizPath)
Visualize paths / Edits (adds/deletes) to paths
Planner Input (Init file)
Contingency Planning Run
Candidate Plan
Visualize Plan
Plan Review
User input for replan
Replan Run
Generate CRL
CRL File
Incorporate tracking info
VAX file with target/landmark info
Go to replan if simulation results rejected
Simulation Run
Simulation Output
Visualize Simulation results
Approve Plan
Process VAX file to create Landmarks file set for Rover
Prepare final plan
Landmarks definition files
Rover Execution of Plan
Data products, Execution from Telemetry
Persist Data Products
Telemetry Data/Exec Trace
Plan exec trace
User Review of Data Products

Assumptions

The following assumptions are relevant to the GDS specification:

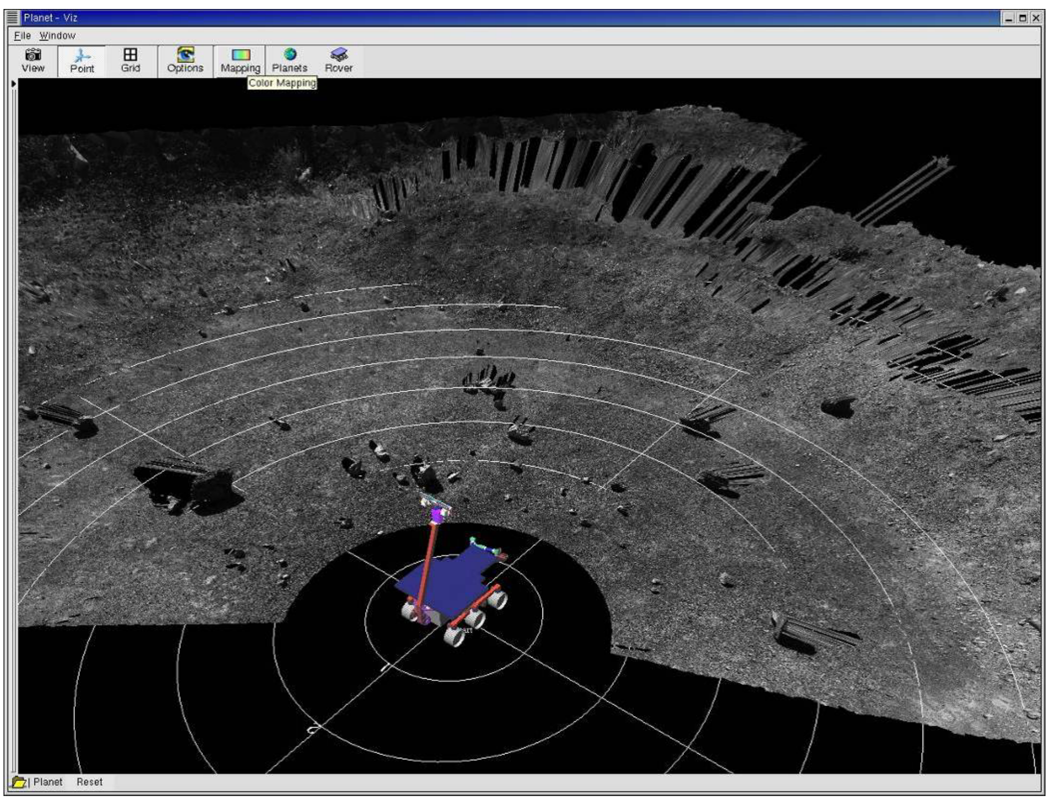
- 1 command cycle / simulated sol
- New panorama available for each command cycle
    - Target selection is restricted to only possible candidates from the latest panorama for the day
- No data volume constraints
- Simulated battery / power for rover
- Orbital imagery for context only
- Instrumentation on board the rover restricted to (a) Microscopic camera (CHAMP) and (b) Hi resolution science stereo cameras

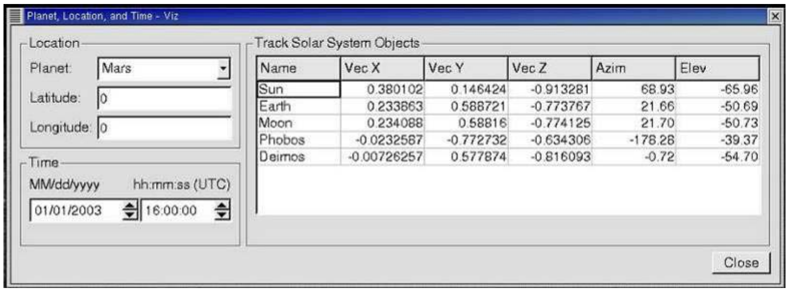
Demonstrations and Results

Date : September 22, 2003	**Location:** NASA Ames Research Center
Goals: User interface demonstration for mission specification and planning	

We present the user-interfaces designed for each of the tasks above

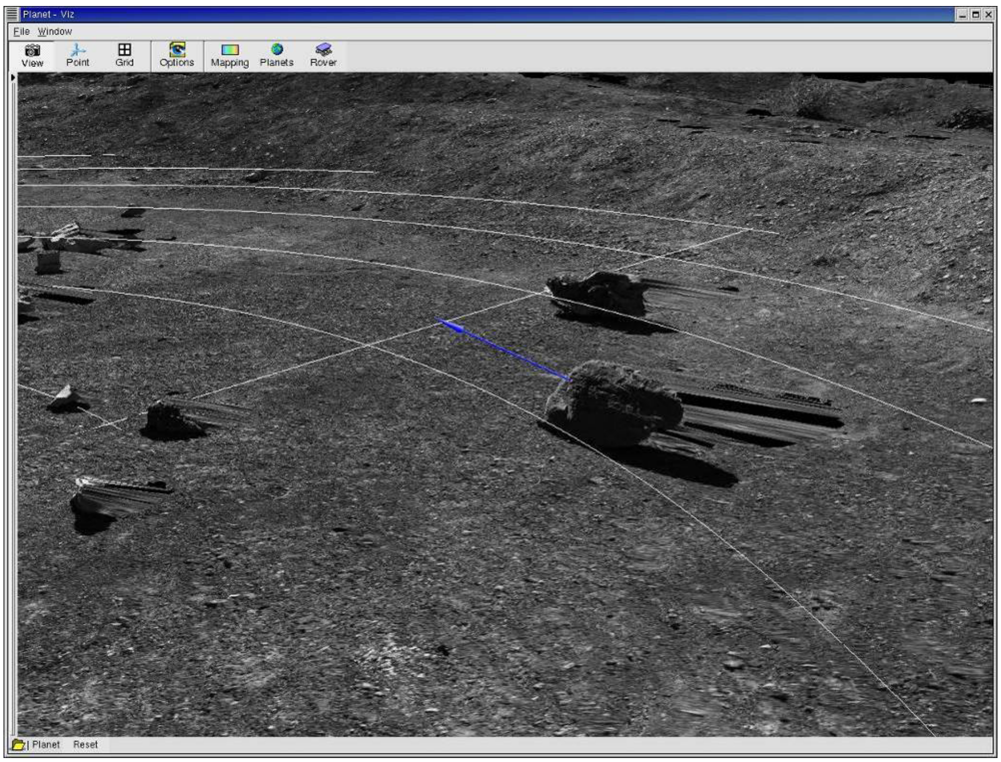
1) Initial Site Assessment (Viz)



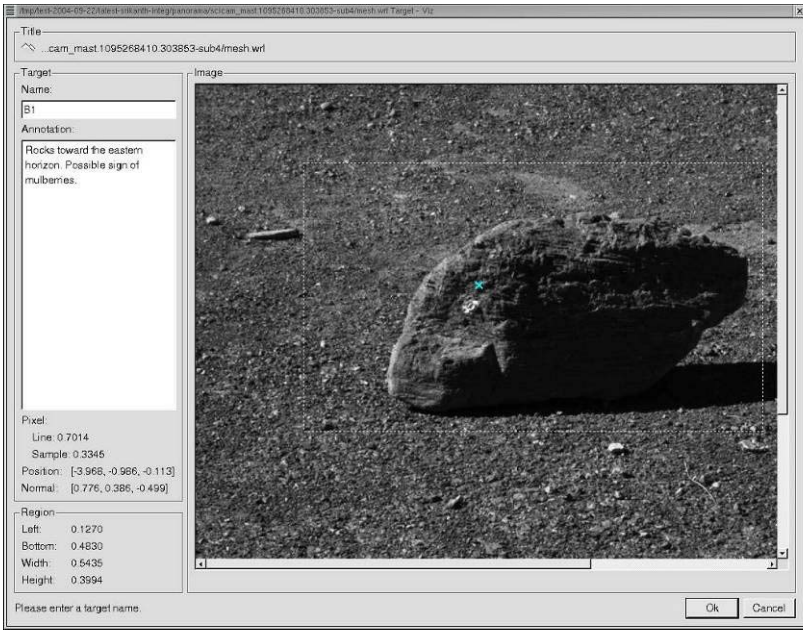


2) Target Requests (Viz)

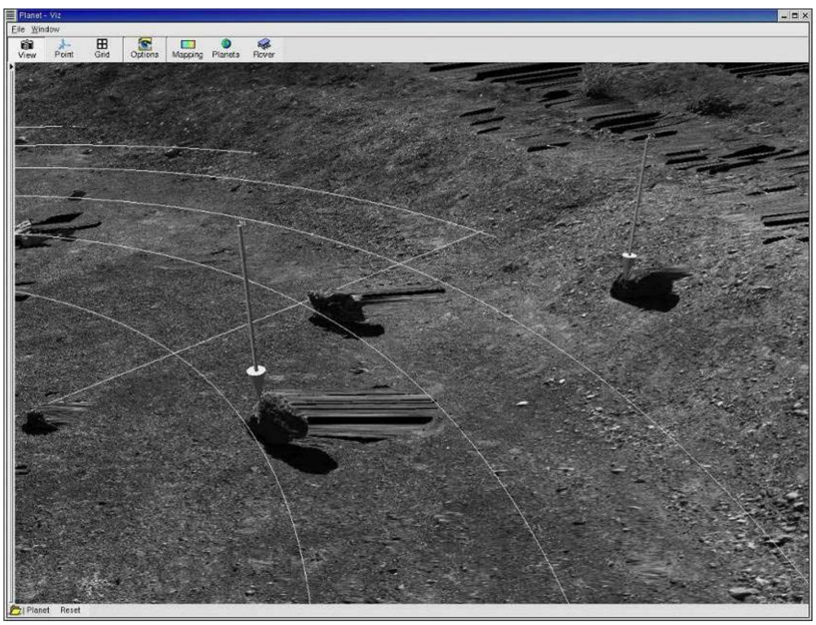
- Users select science target points by clicking on them in 3D model



- User specifies properties of target points:
  - Original camera image of target point presented to user, showing selected location.
  - User indicates rectangle around point, indicating area of model to be used as a tracking template. Viz uses this to generate 3D target template models for use by rover to track targets.
  - User enters target name and additional annotations.



- Target points indicated by 3D arrows in virtual environment.

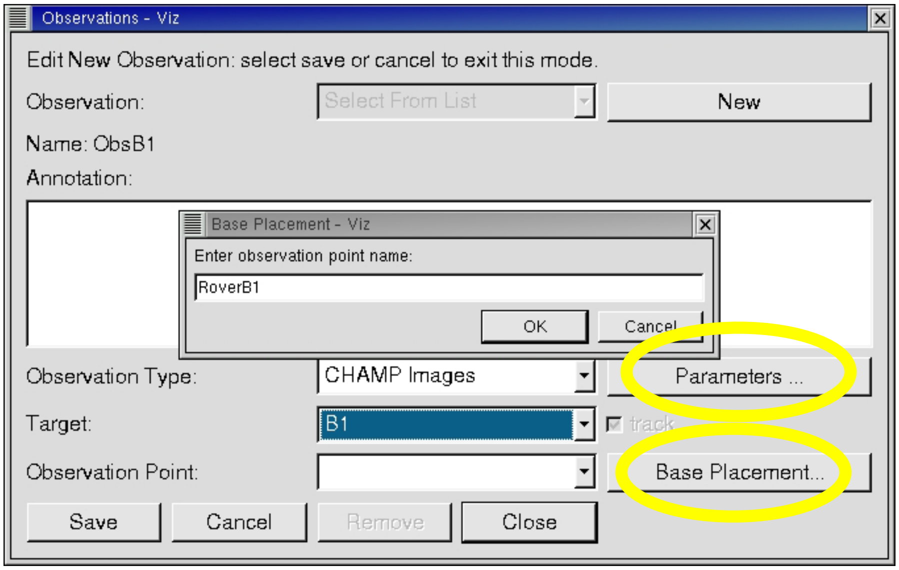


3) Observation Requests (Viz)

For each specified science target, specify requested observations:

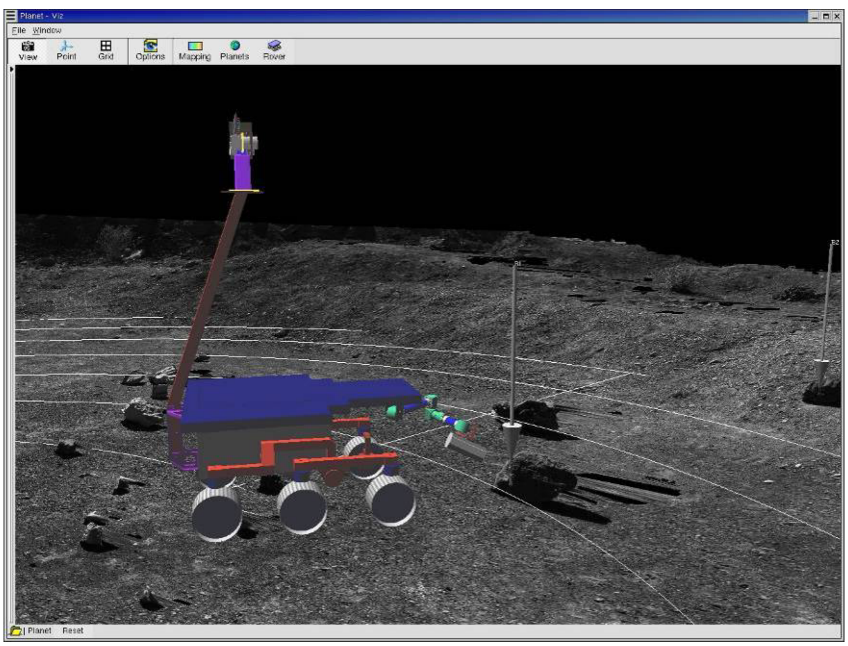
- Instrument
- Instrument parameters (if any)
- Observation Points – where rover must be in order to acquire observation of specified target.

CHAMP Microscopic Image Observation Request:



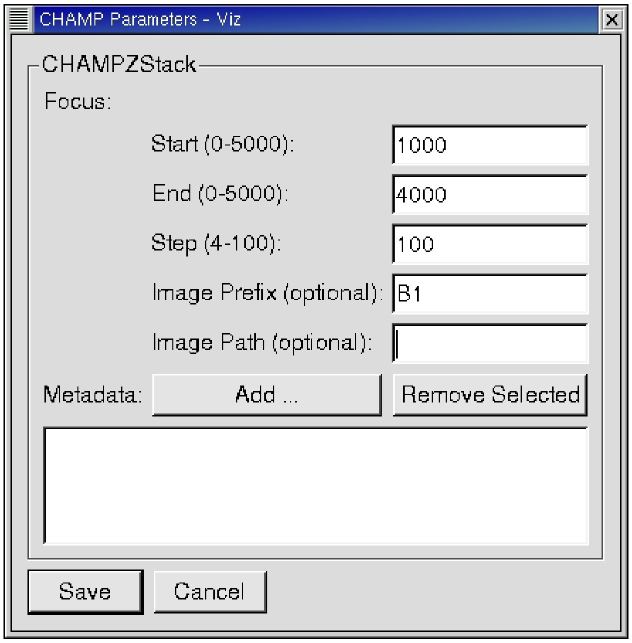
CHAMP is mounted on the rover arm. The rover must be in front of the target rock so that the target instrument pose is in the workspace of the arm.

The automatic Base Placement option above automatically computes the optimal Observation Point and orientation for rover that brings the target point and orientation within the arm workspace. Doing this manually has proven tedious and subject to error.

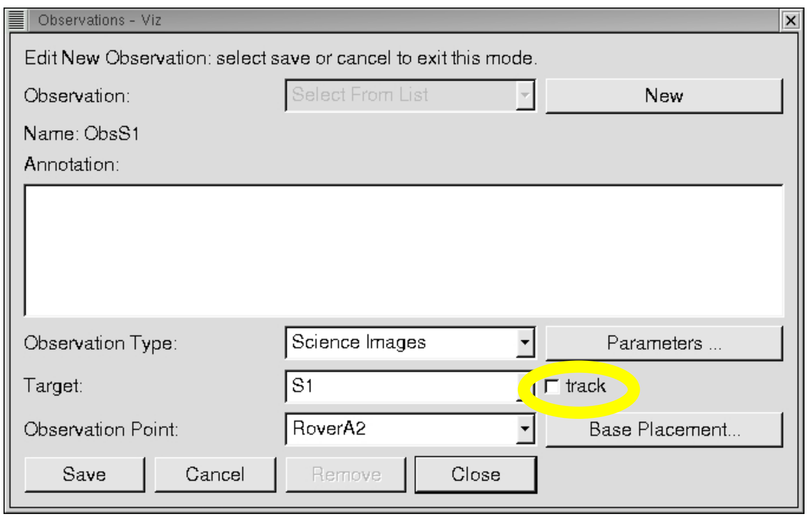


A 3D model of the rover is inserted at the new operations point, for the user to confirm that it is indeed a suitable point, and manually adjust it if not.

Observation parameters:

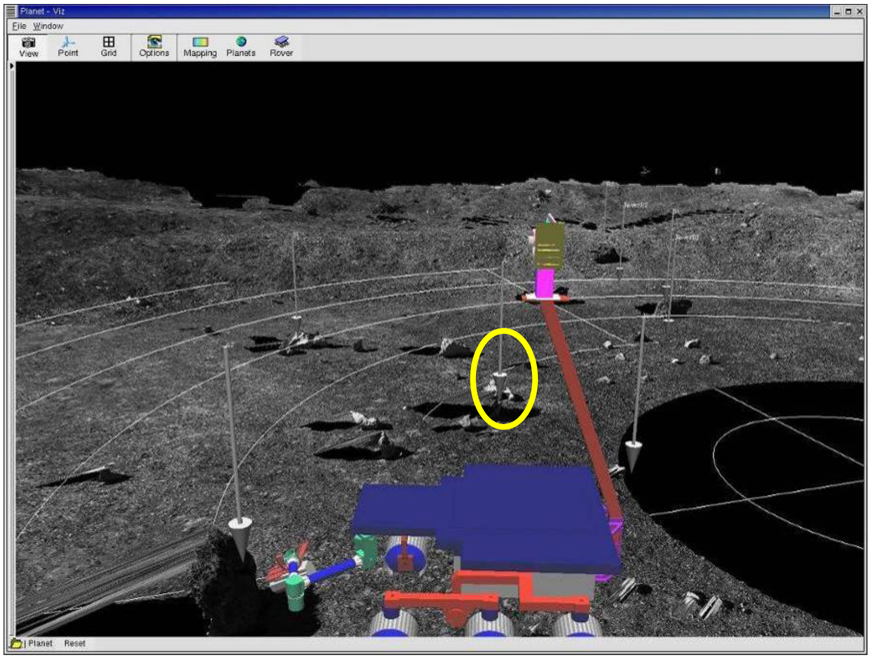


Science Camera Image Requests:

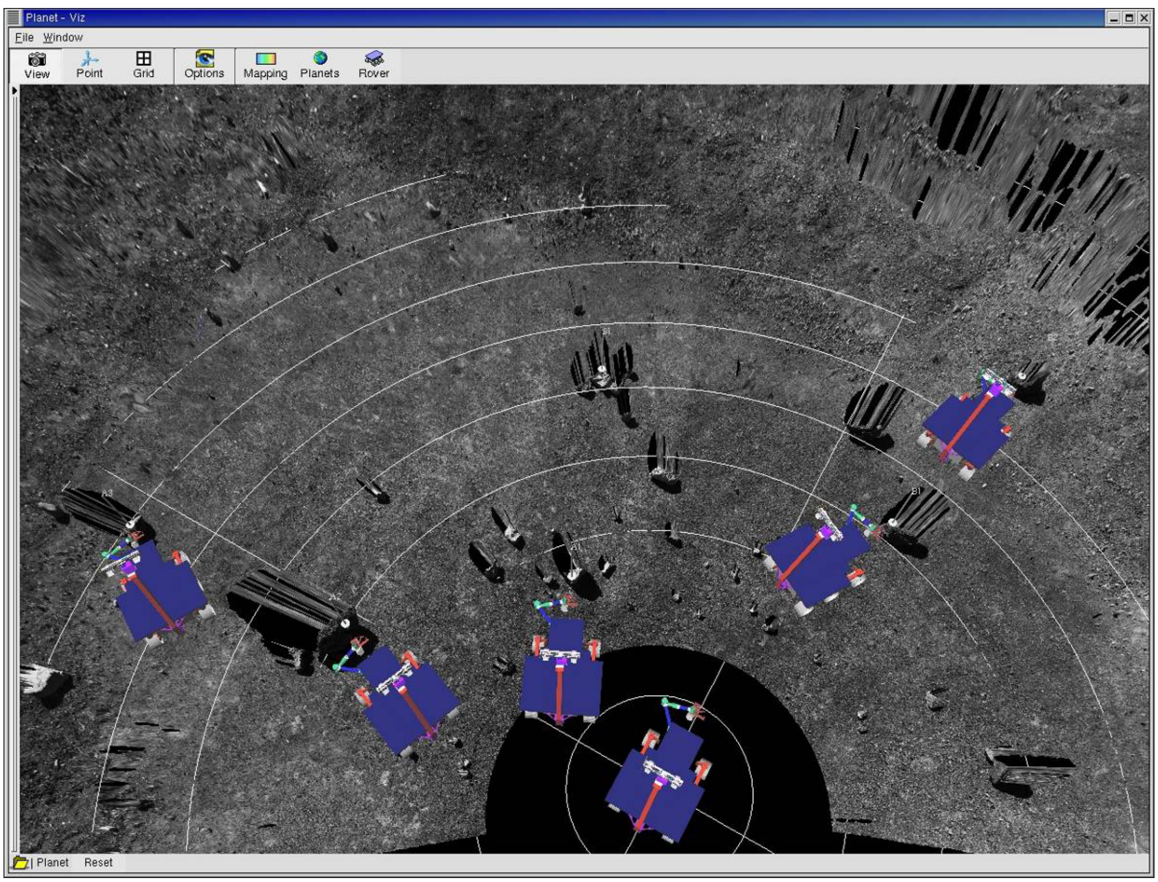


Observation point can be any existing point. Additional observation points can be independently added as needed.

Can specify whether the target should be explicitly tracked or not (target tracking slows rover significantly). This is always the case for a CHAMP (microscopic image), but not necessarily so for science camera images of a general area.

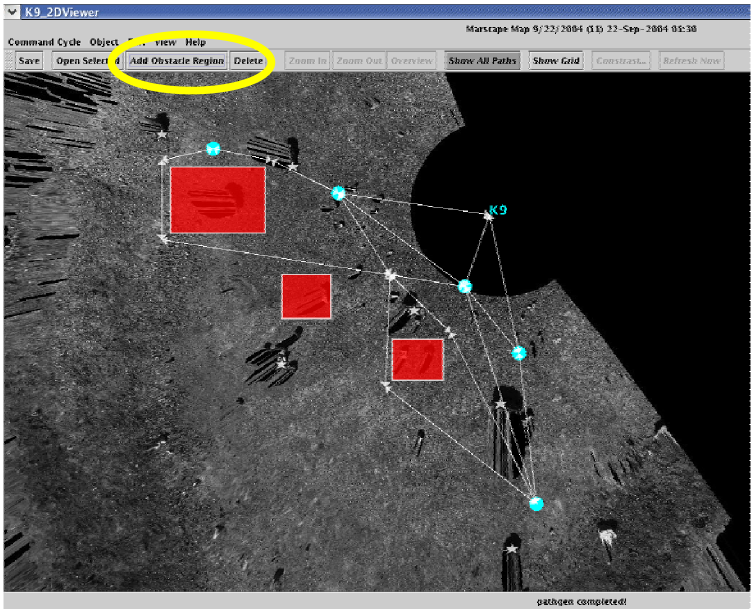


These steps repeated for all targets, as visualized below:

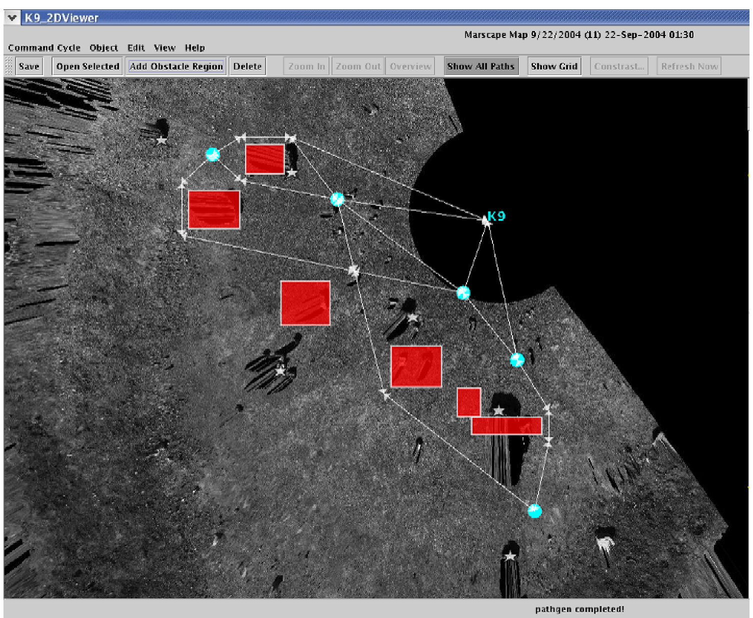


4) Path Specification (PlanView)

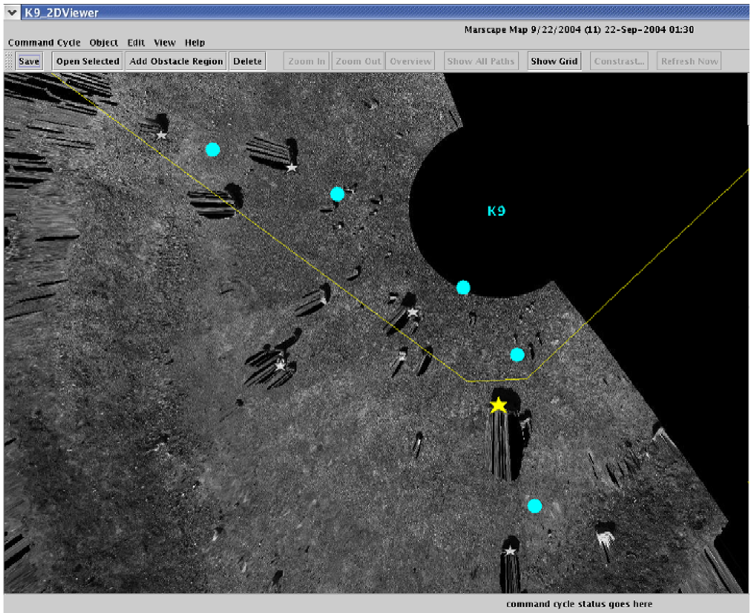
Users specify obstacles as rectangular regions. The *compute paths* button in PlanView sends the obstacle information and target visibility regions (computed by Viz) to the PathGen module, which returns the path network, shown below:



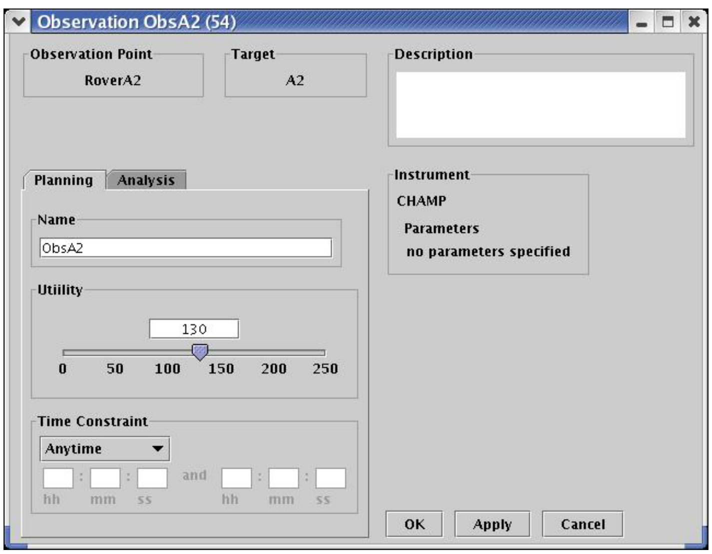
In this case, the returned paths are not safe, crossing a couple of the target rocks. Upon review, the users add more obstacle zones and recomputed paths until satisfied:



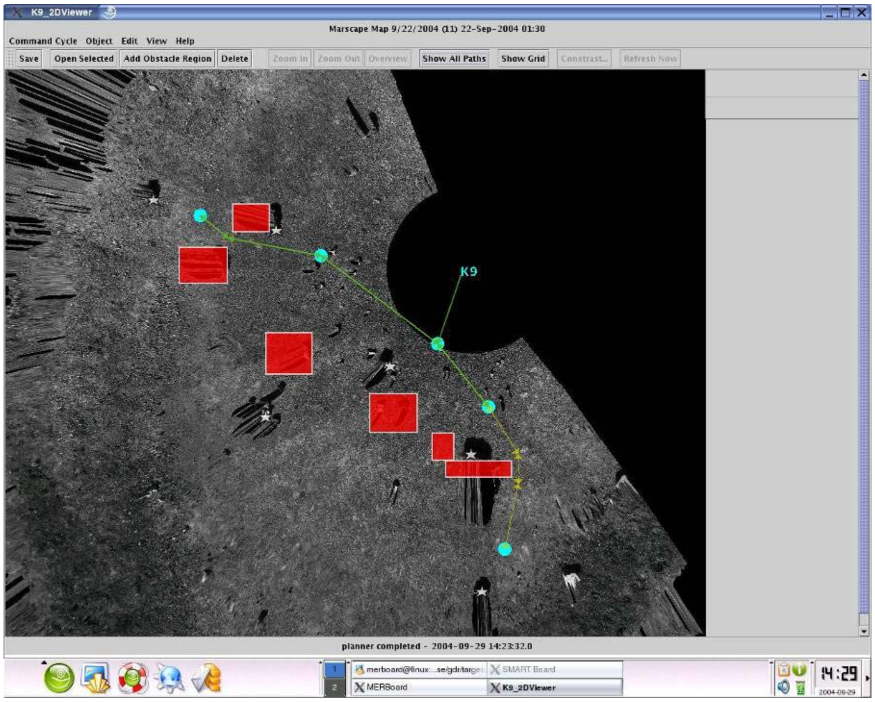
Tracking regions associated associated with a target can be seen in PlanView by clicking on the target icon (star), which becomes highlighted in yellow:



5) Observation Utility and Constraint Specification



6) Activity planning



7) Data Products Visualization

To be demonstrated in FY 2005

Performance

Activity	Time required
Preliminary Site Analysis	As required
Target Requests	~2-3 minutes
Observation Requests	~ 2-5 minutes
Path Specification	10 sec per obstacle 10 sec to generate paths ~ 2-5 minutes overall
Activity Planning	~ up to 10 minutes overall, depending on plan quality. Planner takes up to 3 minutes to generate a new plan.
Data Products Review	TBD

Further Work

- Incorporation of previous days data/analysis into current day's planning/operations – unclear on how much can be achieved here

- Power at the beginning of each day – to provide science/ops team with a view of the range of the rover given power at the beginning of the day?
- Data products review

Simulation-based Technology Development

Goals

In supporting the IS Level 1 Milestone technology development, the Mission Simulation Facility provided several important capabilities.

Researchers in robot autonomy typically focus their resources and expertise on solving a particular problem or developing a specific new approach. Often, research teams do not have time, budget, interest or background in creating software for objective testing. The MSF offers a generic simulation framework intended for technology maturation and mission infusion that is available on a variety of platforms.

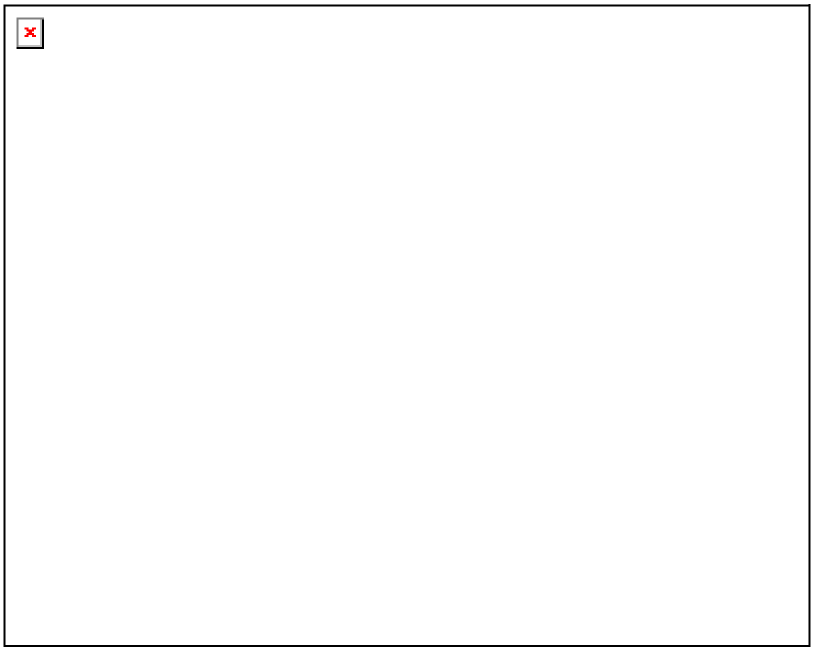
Novel autonomy algorithms often begin with limited capability and grow in sophistication as the technology matures. In early stages of development, autonomy software may not be ready for real-world testing. The MSF can serve as a bridge between overly simplistic test situations, and the dauntingly complex real world by offering a range of simplifications in models of the vehicle, environment, and onboard equipment.

Even for robust autonomy software, field time on real robots is not always the ideal testing approach. Robot platforms tend to be very expensive and in high demand among autonomy researchers. Field test opportunities may be rendered unproductive due to delays and problems unrelated to the autonomous control software intended for testing. The MSF can model many features of actual vehicles and real-world terrain.

Autonomous control software frequently includes branches of reasoning related to hazardous or off-nominal conditions for the robotic vehicle. Executing on a simulated vehicle on virtual terrain offers the opportunity to test portions of code that are difficult to exercise in real situations. Additionally, a readily available simulated vehicle can support numerous repetitions of a test scenario whereas field time on a real rover is very limited.

Technical Approach

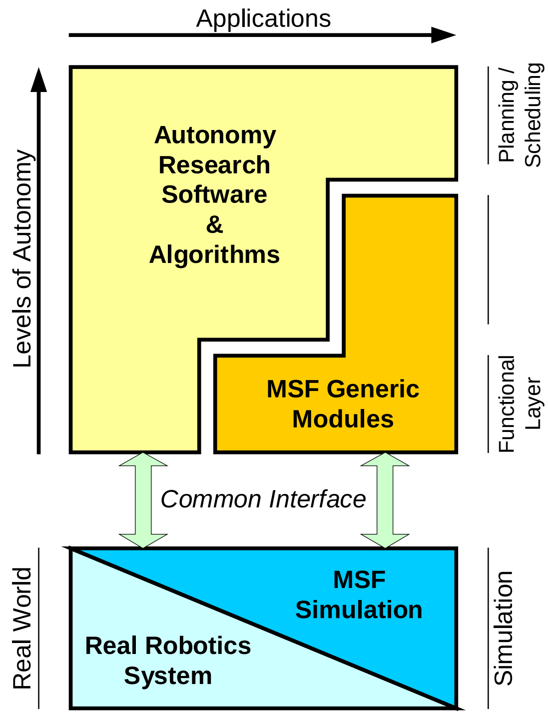
The MSF provides a software framework for the development of autonomous software and autonomy technology for robotic vehicles. The MSF's transport layer is built on HLA (High Level Architecture developed by the Department of Defense) for easy integration of autonomy and simulation modules. Execution can be on single machine or distributed among multiple processors and the system runs on a variety of computer platforms. An example configuration in the following diagram outlines the basic architecture of MSF.



The MSF provides a component based architecture that communicates via a publish and subscribe architecture.

Since the MSF is designed to be flexible, significant development effort focused on generic descriptors and interfaces. A simulated vehicle may contain many conceptual representations in a single application. A vehicle description may include the vehicle's physical characteristics (size, mass properties, configuration), and appearance for graphical display. Another way to characterize a vehicle is with a data dictionary which represents the vehicle's capabilities in terms of input and output as well as the subsystems on board (for example, power supply or independent payload models). Additionally, the model may also include a functional description of the operation of the vehicle that could be understood by an intelligent controller.

Similarly, the virtual environment must be defined in terms relevant to many perspectives: dynamic interactions with the vehicle, appearance in graphical output, characteristics related to sensors and instruments, features that are meaningful to human users of the system, abstract functional descriptions, and any significant changes with time. In the MSF's approach to creating a simulated world, all the user input definitions are maintained in a file structure that eliminates redundant information so that changes made in one place will be reflected throughout the system.



A common, flexible API and multiple levels of abstraction allow the autonomy researcher to easily customize the MSF to address their research focus.

Well-defined interfaces allow interchangeability of real hardware with simulated components. Developers can port their product from the MSF to real platforms without having to maintain separate interfaces. Another advantage of presenting the user with a clean API is easy extensibility to new software elements.

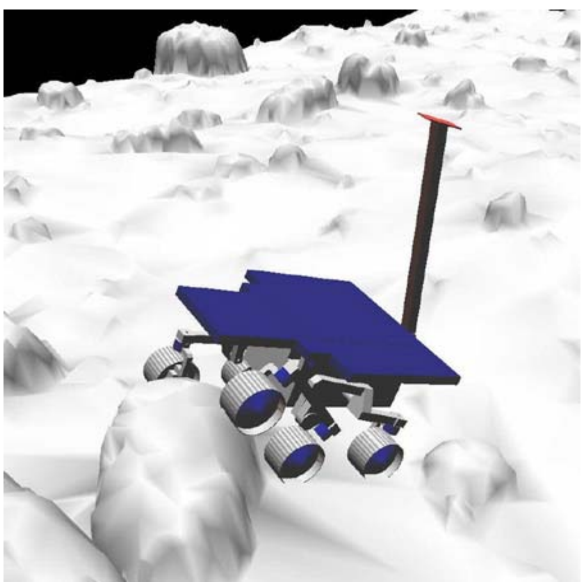
The diagram above illustrates the modular design approach of the MSF, which allows users to customize the simulation to include the layers of abstraction appropriate for the testing situation. For autonomy research, which includes capabilities ranging from abstract planning and scheduling all the way to detailed functional commands, the MSF offers interfaces directly to the robotic platform. In cases where research software focuses on high-level decision making only, the MSF provides intermediate layers of abstraction between the autonomous component and the robot.

The Mission Simulation Facility is a simulation system that represents a diverse collaboration effort. The core technology of the MSF offers a framework for connectivity among modules provided by users or collaborators. Major components of the synthetic world are the terrain surface, environmental conditions, virtual robot, simulated equipment, and graphical display. The MSF design includes technical features essential to support simulation applications. The following describes MSF components, interfaces, and capabilities.

Simulating the ground we walk on is a significant technical challenge. A triangle mesh created from a digital elevation map (DEM) overlaid with realistic textured imaging is a useful structure for portraying a surface in computer graphics and is part of the MSF terrain model. However, researchers in planetary surface robotics require many additional layers of detail. For example, an object-oriented format captures the existence and locations of specific features such as rocks and craters while interaction with scientific instruments is supported with information pertaining to spectrographs and mineral content.

Most of the terrain used in MSF to date comes from collaboration with researchers at NASA's Jet Propulsion Laboratory (JPL). The Simscape project is a server-based provider of artificial, realistic, or real-world terrain data including both physical and science characteristics. Terrain used in MSF simulations can be generated so users can specify characteristics such as rock size and distribution[4]. Specification for artificial terrain may reflect constraints that are exaggerated or over-simplified for specific testing purposes and realistic terrain would be based on knowledge of typical planetary surface conditions. In addition, virtual renditions of real-world sites can be integrated in the MSF database format, allowing simulated robots to drive on synthetic or real-world surfaces. Continuous terrain is available in contiguous patches.

A virtual robotic vehicle offers numerous advantages over real hardware. Depending on the user's research emphasis, autonomy development may be best supported by allowing perfect navigation, instantaneous location changes, unlimited power supply, or perfect sensor readings. In contrast, other researchers may need to introduce navigation errors, locomotor inefficiency, unplanned power shortages, or noisy sensor readings. This screen snapshot illustrates a virtual rover operating on virtual terrain.



Virtual K9 driving over a synthetic rock.

An important feature of the MSF is the capability to provide varying levels of simulation fidelity. With software for high level reasoning, such as planning or resource allocation, the module under development might require only summary status information as input: command completion, for example. In cases involving only high-level abstractions, there is no need to simulate detailed hardware functionality; a simple "stub" will do. In other cases, such as fault diagnosis or science data processing, there may be a need for much higher fidelity in the simulated vehicle and its interactions with the environment.

Assumptions

Level of fidelity. For the IS Level 1 Milestone, MSF supported the development of algorithms which address a high level of abstraction in the representation of vehicle functionality. That is, the autonomous capabilities were not concerned with low-level control laws and hardware drivers onboard the vehicle. The simulated vehicle in MSF is modeled at moderate fidelity. There are vehicle models available which are far more detailed, however their use typically supports engineering simulations of detailed functionality which were not appropriate in this application.

Rover & terrain. For purposes of the IS Level 1 Milestone demonstrations, the Mission Simulation Facility provided models of the K9 rover operating on unchanging Mars-like terrain or the Marscape test site. The Level 1 Milestone demo does not require virtual vehicles representing other rover specifications, such as MER, or virtual test sites representative of complex operations such as mining or habitat construction.

Kinematic modeling. MSF has the capability to model rover movement in terms of kinematics or dynamics. Including force models increases the complexity and processing time for calculating vehicle movement. In the case of a slow-moving planetary exploration rover, and its use for demonstrating autonomy technology, simple models of kinematic movement are more than sufficient to represent the behavior of the vehicle.

Real time execution. The Mission Simulation Facility supports execution speeds equivalent to real time, which provides useful information to autonomy researchers. MSF also supports execution speeds slower and faster than real time for specific research purposes. However, it is beyond the scope of the project to support execution on and embedded processor in true real time.

Demonstrations and Results

Date: January 2004	**Location**: Ames Research Center
Goals: Provide Support for event-driven simulation components **Results**: A multi-threaded simulation loop was implemented to simplify the integration of the MSF component with external applications. This enhancement allowed the smooth and transparent execution of the MSF control and communication component in parallel with the main application loop. This threaded simulation loop was integrated with time management to support initial synchronization, simulation resets, and simulation design. These additional facilities (time query + thread wakeup on timer) which exposed the simulation time to the simulator user enabled more elaborate testing of the Conditional Executive.	

Date: February 2004	**Location**: Ames Research Center

Goals: MSF component integration and support of Planner and Executive integration **Results**: The MSF continued to be enhanced to meet the IS milestone integration. Our HLA browser was enhanced to provide new debugging capabilities including command issuance, execution, and data logging. A K9 tracking module was designed and integrated. This was used as a stand in for the integration team while the real K9 tracking system requirements were defined.

Date: June 2004	**Location**: Ames Research Center
Goals: Support of Engineering Readiness Test 2 (ERT2) **Results**: The MSF participation in the IS ERT2 was highly successful. .In addition to adding new features during the test the MSF was able to support all the required scenarios. The Conditional Executive developers used the MSF system to test and demonstrate that its component could provide all the capabilities required for ERT2. While initially testing the code on the MSF simulator, a race condition in was discovered in the Conditional Executive code that may not have been exercised in the full robotic test. .In addition to adding new features during the test the MSF was able to support all the required scenarios. The following components participated in the MSF simulation: the Conditional Executive, ROAMS rover, Rock Detector, Target Tracking and K9-Subsystems (including instrument placement and power model), and the Ames Marscape terrain model populated with rocks that can be repositioned.	

Date: August 2004	**Location**: Ames Research Center
Goals: Support of Operational Readiness Test 2 (ORT2) **Results**: The MSF was used to support the tests of increased functionality of the planner using modeled tracking regions and more elaborate branching.	

Performance

The most effective improvements in the technology development process due to the use of a simulation testing environment are qualitative. Opportunities for autonomy researchers to test their code on a rover test platform have customarily been infrequent. Visual displays offer rich information to the programmer for assessing software behavior and MSF users found inestimable value in the ability to *see* their code execute. MSF also records data and command messages for later playback or analysis.

The capability to execute all branches of software is available in simulation but risky on real hardware. During technology development for the IS Level 1 Milestone, simulation testing discovered a bug that probably would never have become apparent with real hardware test runs.

The most quantitative assessment of the simulation-based development approach addresses the number of test runs executed. In simulation, many comparison runs can be executed on the course of an hour, where real test hardware carries inherent difficulties of low repetitions, overhead of transport and set up time, and down time due to weather, breakage, or other research uses.

References

G. Pisanich, L. Plice, C. Neukom, L. Flueckiger and M. Wagner, Mission Simulation Facility: Simulation Support for Autonomy Development, AIAA-2004-947, 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, Jan. 5-8, 2004

Pisanich, G., Flueckiger, L., Neukom, C. A Facility and Architecture for Autonomous Research, *In Proceedings of IITSEC Conference*, Orlando FL. December 2002

L. Flückiger and C. Neukom. A new simulation framework for autonomy in robotic missions. *In Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, EPFL, Lausanne, Switzerland (pp 3030 -- 3035) June 2002.

L. Edwards, L. Flückiger, and R. Washington. VIPER: Virtual intelligent planetary exploration rover. In *Proceedings of i-SAIRAS 2001*, 2001.

System Demonstrations

<table>
<tr><td>Date : October 2003</td><td>Location: GraniteRock Aromas quarry, Watsonville, CA and NASA Ames Research Center</td></tr>
<tr><td colspan="2">Goals: End-to-end demonstration of ground operations, contingent planning, the crl executive, science autonomy, navigation and instrument placement

Results: Videos\ERT1-2003.mov</td></tr>
</table>

- Operators at NASA ARC designated 2 targets in Viz
- An early version of PICo, with humans in the loop to choose branch points, was used to generate a plan to visit on of the 2 targets, branching on energy.
- The CRL plan was uplinked via satellite to the field location and executed on K9.
- K9 tracked both targets using mesh registration and placed CHAMP on one of them as dictated by the plan.
- During the traverse, science autonomy routines detected layers on a nearby rock, triggering a floating contingency that directed the rover to acquire hi-resolution follow up images of the target.

Date : Fall 2004	**Location:** NASA Ames Research Center Marscape and Mission Ops Center
Goals: A public demonstration of the integrated single cycle instrument placement technologies, as described in this document, showing the overall performance goals advertised in this document.	

This demonstration is planned subsequent to the release of this document.

Acknowledgements

This work was supported by the Intelligent Systems (IS) Project, with significant contributions from the Astrobiology Science and Technology for Exploring Planets (ASTEP) and Mars Technology Development (MTD) programs.

Special thanks to the following, whose support was essential for this project:

Butler Hine

Robert A. Morris

James Crawford

Nicola Muscettola

Daniel Clancy

Critical Design Reviewers:

Hans Thomas

Steve Rock

David Wettergreen

Steve Peters

Paul Schenker

EUROPA/PLASMA Development Team:

Conor McGann

Tania Bedrax-Weiss

Ari Jonsson

Media Coverage

KPIX-TV Ch 5 CBS (San Francisco, USA, network TV), KNTV-Ch 11 NBC (San Jose, USA, network TV), stories on K9 rover demonstrations, December 2002. Press release, demo footage materials and interviews.

KLIV News Radio(San Jose, USA, radio), KWZ Radio (Cincinnati, USA, radio), WVXU-FM NPR Cincinnati (Cincinatti, USA, radio), January 2003. Press release and interviews (by phone).

Tech TV (USA, TV program) – February 2003

Telemundo TV (USA, network TV) - February 2003

And more….

Publications

Movies

Pedersen, L., "Autonomous Instrument Placement for Mars Rovers", in *IEEE International Conference on Robotics and Automation 2003,* Taipei, Taiwan, 2003.

Conference Proceedings

Deans, M., C. Kunz, R. Sargent, L. Pedersen, "Terrain Model Registration for Single Cycle Instrument Placement," in *International Conference on Intelligent Robots and Systems (IROS),* Las Vegas, USA, 2003.

Pedersen, L., M. Bualat, C. Kunz, S. Lee, R. Sargent, R. Washington, A. Wright "Instrument Placement for Mars Rovers", in *IEEE International Conference on Robotics and Automation 2003,* Taipei, Taiwan, 2003.

Pedersen, L., R. Sargent, "Single Cycle Instrument Deployment for Mars Rovers", in *International Symposium on Artificial Intelligence and Robotics in Space (i-SAIRAS) 2003,* Nara, Japan, 2003.

Pedersen, L., M. Bualat, D.E. Smith, R. Washington, "Integrated Demonstration of Instrument Placement, Robust Execution and Contingent Planning", in *International Symposium on Artificial Intelligence and Robotics in Space (i-SAIRAS) 2003,* Nara, Japan, 2003.

Deans, M., R. Sargent, C. Kunz, L. Pedersen, "Terrain Model Registration for Navigation and Effector Control", in *International Symposium on Artificial Intelligence and Robotics in Space (i-SAIRAS) 2003,* Nara, Japan, 2003.

Pedersen, L., "Science Target Assessment for Mars Rover Instrument Deployment", in *International Conference on Intelligent Robots and Systems (IROS),* Lausanne, Switzerland, 2002.

More publications in progress following public demonstration and recent results.

REVISION CONTROL

Rev Number	Date	Owner	Comments
1.0	September 25, 2004	Liam Pedersen	First draft
1.1	September 30, 2004	Liam Pedersen	Comprehensive draft with technology sections re-organized, new material on planner, ip, keypoint tracking results.